

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC VÀ KỸ THUẬT THÔNG TIN



BÁO CÁO ĐỒ ÁN
MÔN KỸ THUẬT LẬP TRÌNH PYTHON

Đề tài:

**SỬ DỤNG CÁC CÔNG CỤ PHÂN TÍCH VÀ
TRỰC QUAN HÓA DỮ LIỆU CỦA PYTHON
ĐỂ PHÂN TÍCH DỮ LIỆU CHO BÀI TOÁN
TRONG CUỘC THI “VINBIGDATA CHEST X-RAY
ABNORMALITIES DETECTION”.**

GVHD: ThS. Nguyễn Thanh Sơn

Nhóm sinh viên thực hiện:

- | | |
|--------------------|----------------|
| 1. Nguyễn Văn Khoa | MSSV: 18520929 |
| 2. Phạm Nhật Dương | MSSV: 18520650 |

🌀 Tp. Hồ Chí Minh, 5/2021 🌀

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

[illegible]

....., ngày.....tháng.....năm 2020

Người nhận xét

(Ký tên và ghi rõ họ tên)

BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN

Họ và tên	MSSV	Phân công	Đánh giá
Nguyễn Văn Khoa	18520929	<ul style="list-style-type: none"> - Tìm hiểu cách sử dụng cơ bản của thư viện pandas và plotly. - Tìm hiểu thông tin nền tảng cùng kiến thức chuyên ngành (domain knowlede) liên quan đến các căn bệnh thuộc 14 class của bài toán trong cuộc thi để phục vụ quá trình phân tích. - Dùng pandas và plotly vẽ các đồ thị để thăm dò từng cột của dữ liệu. - Truyền đạt lại các bước làm cũng như giải thích những khúc mắc cho cộng sự. - Quản lý phiên bản và cập nhật github. - Viết báo cáo đồ án. 	Hoàn thành
Phạm Nhật Dương	18520650	<ul style="list-style-type: none"> - Làm quen cách sử dụng cơ bản của thư viện pandas và matplotlib. - Thực hiện 2 project nhỏ để chạy, hiểu và giải thích các câu lệnh của pandas và matplotlib trong project đó. - Tìm hiểu thông tin nền tảng của bài toán chính trong cuộc thi. - Dựa theo cách làm của cộng sự để học và quen dần với quy trình của một dự án phân tích dữ liệu thăm dò thường thấy. - Chạy lại code theo hướng dẫn để quen với qui trình. - Viết báo cáo đồ án 	Hoàn thành

Bảng 1: Bảng phân công đánh giá thành viên

MỤC LỤC

BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN.....	3
LỜI MỞ ĐẦU	5
Chương 1: THÔNG TIN CHUNG VỀ BÀI TOÁN TRONG CUỘC THI. 7	
Chương 2: THĂM DÒ CỘT IMAGE_ID	10
Chương 3: THĂM DÒ CỘT CLASS_NAME	16
Chương 4: THĂM DÒ CỘT CLASS_ID	17
Chương 5: THĂM DÒ CỘT CLASS_ID	18
Chương 6: THĂM DÒ CÁC CỘT TỌA ĐỘ BOUNDING BOX	20
Chương 7: CÁC TRỰC QUAN HÓA DỮ LIỆU KHÁC	21
TÀI LIỆU THAM KHẢO	25

LỜI MỞ ĐẦU

Trong hầu hết các bài toán thuộc lĩnh vực Khoa học Dữ liệu (Data Science) mà cần phải dùng các phương pháp đến từ Học Máy (Machine Learning) hay Học Sâu (Deep Learning) để giải quyết, chúng ta đều phải làm việc với Dữ liệu Lớn (Big Data). Cho dù đó là bài toán liên quan đến Thị giác Máy tính (Computer Vision - CV) hay Xử lý Ngôn ngữ Tự nhiên (Natural Language Processing - NLP) hay các lĩnh vực khác.

Thông thường dữ liệu này được thu thập theo cách ghi lại nhiều thông tin nhất có thể, tức là quá trình thu thập dữ liệu sẽ khó mà mang đến một bộ dữ liệu tối ưu và “sạch sẽ” đến mức có thể sử dụng ngay từ đầu. Nguyên nhân có thể đến từ những sai lệch trong quá trình thu thập hay đến từ đặc thù của cách thu thập dữ liệu trong bài toán. Do đó, quy trình thường thấy của một dự án Data Science sẽ có các bước đầu tiên là: (1) Xác định bài toán, (2) Chuẩn bị dữ liệu, (3) Phân tích và trực quan hóa dữ liệu, (4) Tiền xử lý dữ liệu, sau đó mới tới bước đào tạo mô hình và các bước sau nó.

Python cung cấp cho chúng ta những công cụ mạnh mẽ để thực hiện tác vụ phân tích và trực quan hóa dữ liệu, tiêu biểu nhất trong số những công cụ đó chính là Pandas và Matplotlib. Pandas là thư viện mạnh mẽ dùng để thao tác và phân tích dữ liệu, nó được cung cấp các cấu trúc dữ liệu và các phép toán để thao tác với các dataframe một cách nhanh chóng. Matplotlib là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy, tuy nhiên gần đây có một thư viện vẽ đồ thị khác đang nhận được nhiều sự chú ý đến từ cộng đồng đó là Plotly. Thư viện Plotly cho phép người dùng tạo ra các đồ thị có thể tương tác được, theo đó người dùng có thể thực hiện các tác động cơ bản lên đồ thị gốc để có một đồ thị mới thể hiện cụ thể hơn một phần nội dung nào đó, điều này thực sự tiện lợi khi ta không cần phải vẽ một đồ thị mới để thực hiện mong muốn đó.

Khác với các giai đoạn khác trong quy trình của một dự án Data Science, để thực hiện được tác vụ phân tích & trực quan hóa dữ liệu, người thực hiện ngoài việc phải nắm rõ các công cụ trợ giúp thì phải có một lượng kiến thức và hiểu biết nhất định về bài toán mình đang xử lý. Lấy một ví dụ như là một bài toán liên quan đến dữ liệu y sinh, khi phân tích dữ liệu dạng này không thể áp dụng một cách máy móc các biểu đồ phổ biến hay các cách phân tích thông dụng lên dữ liệu, người thực hiện bắt buộc phải có

cái nhìn sâu sắc về từng căn bệnh, từ đó có được hướng phân tích và trừu tượng hóa hợp lí.

Để làm rõ hơn về vấn đề này cũng như rèn luyện kỹ thuật lập trình Python của mình trong việc phân tích dữ liệu, trong đồ án này, sinh viên sẽ sử dụng hai thư viện chính là pandas và plotly để thực hiện phân tích dữ liệu thăm dò trên dữ liệu của bài toán trong cuộc thi “VinBigData Chest X-ray Abnormalities Detection”. Cụ thể, mục tiêu đặt ra của đồ án như sau:

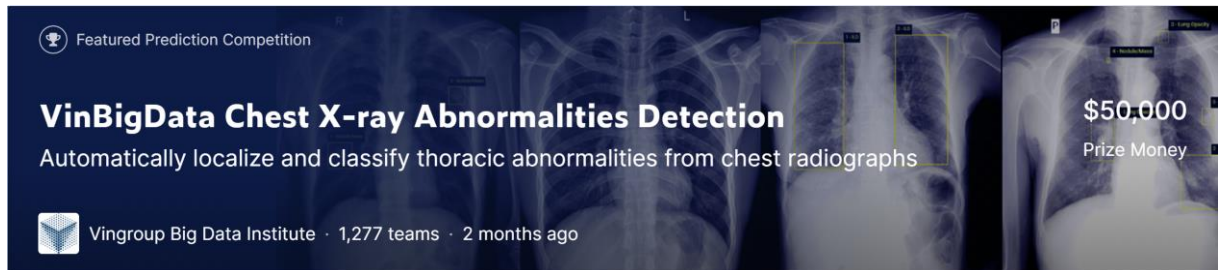
- Tìm hiểu cách sử dụng cơ bản của pandas và plotly
- Tìm hiểu về bài toán mà mình đang xử lý
- Thực hiện khai thác từng cột của tập dữ liệu để phân tích những điểm quan trọng trong bộ dữ liệu thông qua việc truy vấn dữ liệu và vẽ đồ thị trực quan.

Để dễ dàng theo dõi và làm rõ các thông tin, bài báo cáo sẽ được chia thành các phần như sau:

- **Chương 1:** Thông tin chung về bài toán trong cuộc thi.
- **Chương 2:** Thăm dò cột image_id.
- **Chương 3:** Thăm dò cột class_name.
- **Chương 4:** Thăm dò cột class_id.
- **Chương 5:** Thăm dò cột rad_id.
- **Chương 6:** Thăm dò nhóm cột thể hiện tọa độ boundingbox.
- **Chương 7:** Các sự trực quan hóa dữ liệu khác.

Chương 1: THÔNG TIN CHUNG VỀ BÀI TOÁN TRONG CUỘC THI.

Trong cuộc thi này, chúng ta sẽ phát hiện những bất thường, những căn bệnh phổ biến trên phổi. Đây là một bài toán Object Detection.



Hình 1.1: Ảnh bìa cuộc thi

Chúng ta sẽ được cung cấp một tập dữ liệu gồm 18000 ảnh chụp X-Quang ở định dạng DICOM. Tất cả những bức ảnh này đều đã được gán nhãn bởi các bác sĩ X-Quang đầy kinh nghiệm, có tất cả 14 căn bệnh như sau:

- Aortic enlargement - Phình động mạch chủ
- Atelectasis - Phù phổi
- Calcification - Vôi hóa phổi
- Cardiomegaly - Tim to
- Consolidation - Đông đặc phổi
- ILD
- Infiltration - Thâm nhiễm phổi
- Lung Opacity - Đục phổi
- Nodule/Mass - U/Bướu
- Other lesion - Các căn bệnh khác
- Pleural effusion - Tràn dịch màng phổi
- Pleural thickening - Phổi dày
- Pneumothorax - Tràn khí phổi
- Pulmonary fibrosis - Thâm nhiễm phổi
- "No finding" Không tìm thấy căn bệnh nào trên phổi

Lưu ý rằng trong bài toán lần này, chúng ta sẽ làm việc với ground truth từ nhiều bác sĩ cùng lúc trên một bức ảnh. Cụ thể là tối ta sẽ có 3 bác sĩ cùng gán nhãn trên cùng một bức ảnh

Các file csv được cung cấp bao gồm:

- **train.csv** - metadata của tập train, mỗi dòng là một object (một ảnh có thể có nhiều dòng)
- **sample_submission.csv** - một file submission mẫu - trong khuôn khổ đồ án này, chúng ta không quan tâm đến file này.

Sau khi đọc file train.csv, ta có kết quả như sau:

TRAIN DATAFRAME

	image_id	class_name	class_id	rad_id	x_min	y_min	x_max	y_max
0	50a418190bc3fb1ef1633bf9678929b3	No finding	14	R11	NaN	NaN	NaN	NaN
1	21a10246a5ec7af151081d0cd6d65dc9	No finding	14	R7	NaN	NaN	NaN	NaN
2	9a5094b2563a1ef3ff50dc5c7ff71345	Cardiomegaly	3	R10	691.0	1375.0	1653.0	1831.0
3	051132a778e61a86eb147c7c6f564dfe	Aortic enlargement	0	R10	1264.0	743.0	1611.0	1019.0
4	063319de25ce7edb9b1c6b8881290140	No finding	14	R10	NaN	NaN	NaN	NaN
5	1c32170b4af4ce1a3030eb8167753b06	Pleural thickening	11	R9	627.0	357.0	947.0	433.0
6	0c7a38f293d5f5e4846aa4ca6db4daf1	ILD	5	R17	1347.0	245.0	2188.0	2169.0
7	47ed17dcb2cbeec15182ed335a8b5a9e	Nodule/Mass	8	R9	557.0	2352.0	675.0	2484.0
8	d3637a1935a905b3c326af31389cb846	Aortic enlargement	0	R10	1329.0	743.0	1521.0	958.0
9	afb6230703512afc370f236e8fe98806	Pulmonary fibrosis	13	R9	1857.0	1607.0	2126.0	2036.0

Hình 1.2: Train dataframe

Tiếp đó, chúng ta thực hiện xem các thông kê cơ bản của tập train:

```
[11]: train_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67914 entries, 0 to 67913
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  -
0   image_id    67914 non-null  object
1   class_name  67914 non-null  object
2   class_id    67914 non-null  int64
3   rad_id      67914 non-null  object
4   x_min       36096 non-null  float64
5   y_min       36096 non-null  float64
6   x_max       36096 non-null  float64
7   y_max       36096 non-null  float64
dtypes: float64(4), int64(1), object(3)
memory usage: 4.1+ MB
```

Hình 1.3: Các thông kê cơ bản của train_df

Chương 2: THĂM DÒ CỘT IMAGE_ID

Cột image_id chứa một Unique Identifier (UID) duy nhất cho biết bounding box đang xét tương ứng với bệnh nhân nào. Bởi vì có 3 bác sĩ cùng gán nhãn cho cùng một bức ảnh và có khả năng là nhiều bounding box khác nhau nên một UID có thể xuất hiện nhiều lần. Tuy nhiên xin lưu ý rằng mỗi UID chỉ tương ứng với một bệnh nhân.

2.1. THỐNG KÊ TỔNG SỐ BOUNDING BOX TRÊN MỖI BỨC ẢNH

```
[17]: # Đầu tiên là tạo câu truy vấn pandas
train_df.image_id

[17]: 0      50a418190bc3fb1ef1633bf9678929b3
      1      21a10246a5ec7af151081d0cd6d65dc9
      2      9a5094b2563a1ef3ff50dc5c7ff71345
      3      051132a778e61a86eb147c7c6f564dfe
      4      063319de25ce7edb9b1c6b8881290140
      ...
      67909  936fd5cff1c058d39817a08f58b72cae
      67910  ca7e72954550eeb610fe22bf0244b7fa
      67911  aa17d5312a0fb4a2939436abca7f9579
      67912  4b56bc6d22b192f075f13231419dfcc8
      67913  5e272e3adbdaafb07a7e84a9e62b1a4c
      Name: image_id, Length: 67914, dtype: object
```

Hình 2.1: Chọn cột image_id

Ở đây chúng ta có 67913 records tương ứng với 67913 UID. Vì mỗi record sẽ tương ứng với một bounding box nên khi ta đếm số UID trùng nhau thì đồng thời chúng ta đang thực hiện đếm số bounding box trên mỗi UID.

Tiếp đó, chúng ta sẽ dùng phương thức value_counts() để thống kê số lần xuất hiện của các UID:

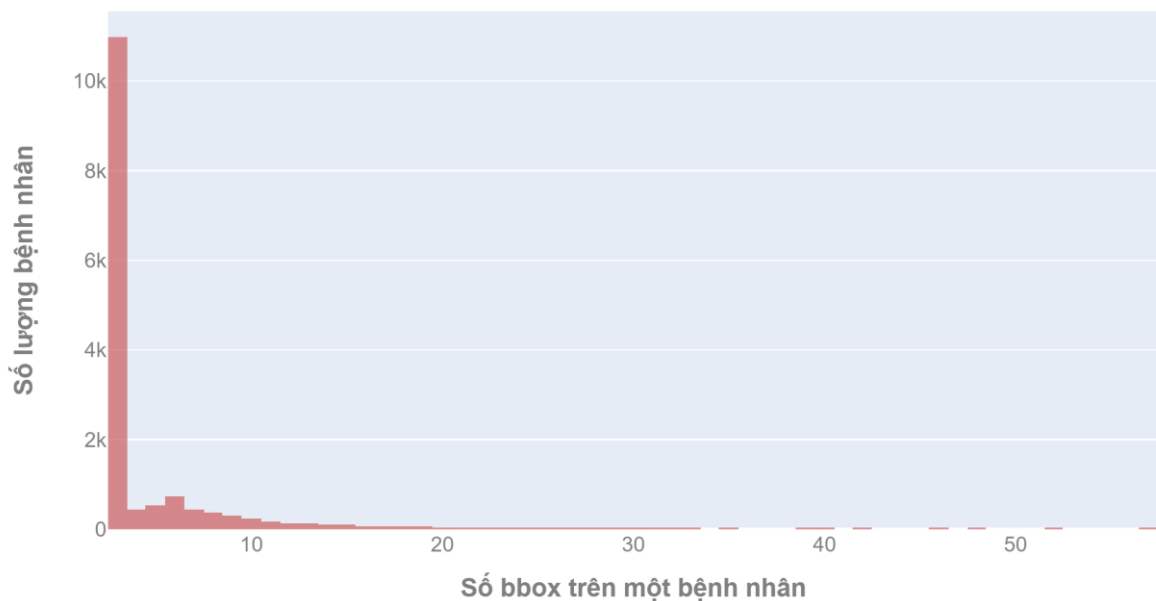
```
[18]: # Dùng phương thức value_counts() để thống kê số lần xuất hiện của các UID
train_df.image_id.value_counts()

[18]: 03e6ecfa6f6fb33dfeac6ca4f9b459c9      57
      fa109c087e46fe1ea27e48ce6d154d2f      52
      e31be972e181987a8600a8700c1ebe88      48
      6d5acf3f8a973a26844d617fffe72998      46
      3a302fbbbf3364aa1a7731b59e6b98ec      46
      ..
      278e688b4ae36a86f24a4f90d2b5e747      3
      a66aa5583aaebc075eaf73317da6557f      3
      f458843968166ba8c5af491035ce1651      3
      ad38cbf4a2b1e6fcb9d78fb513788c      3
      3c47131a874576cb08b30dae80e5eb13      3
      Name: image_id, Length: 15000, dtype: int64
```

Hình 2.2: thống kê số lần xuất hiện của mỗi UID

Từ thông tin này chúng ta đã có thể vẽ biểu đồ:

PHÂN PHỐI CỦA SỐ LƯỢNG BBOX TRÊN MỖI BỆNH NHÂN (Không Log Scale trục y)



Hình 2.3: Biểu đồ thể hiện phân phối của số lượng bounding box trên mỗi bệnh nhân

Từ biểu đồ trên, có thể rút ra được những thông tin sau:

- Một ảnh sẽ có thể có tối thiểu là 3 bounding box (1 căn bệnh duy nhất được gán nhãn bởi 3 bác sĩ x-quang)
- Một ảnh sẽ có thể có tối đa là 57 bounding box (19 căn bệnh riêng biệt được gán nhãn bởi 3 bác sĩ x-quang)
- Phần lớn các ảnh chỉ có ba bounding box (~11,000 trên tổng số 15,000)
- Phân phối có một độ lệch rất lớn

Cụ thể độ lệch sẽ được tính như sau:

```
[21]: scipy.stats.skew(train_df.image_id.value_counts().values)
```

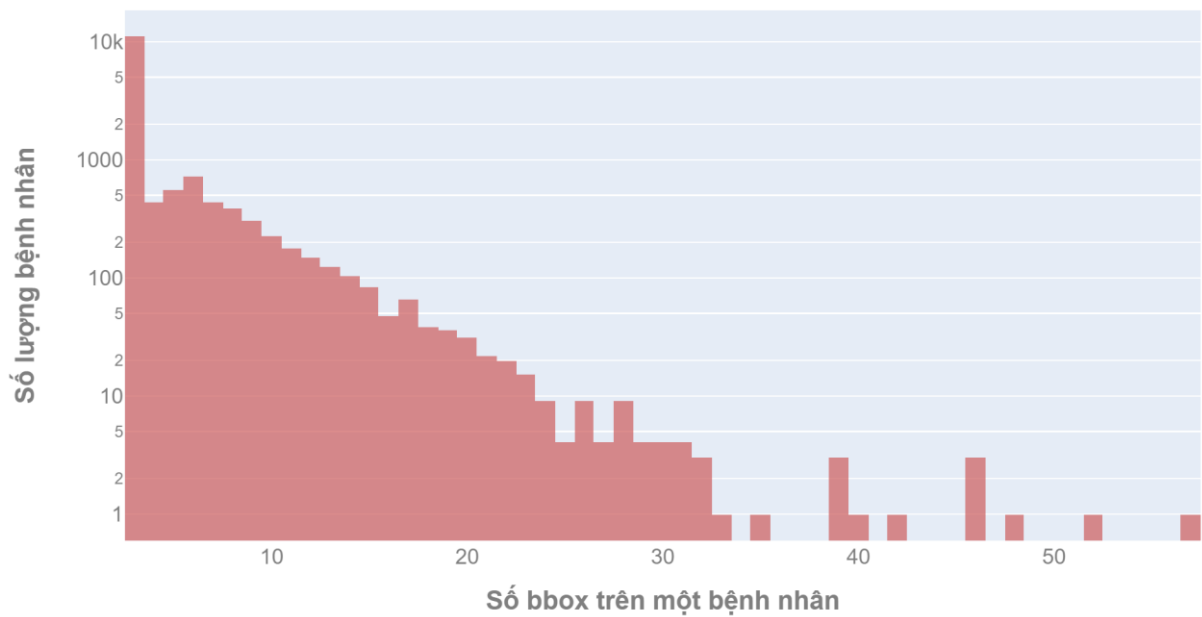
```
[21]: 3.8687405565463298
```

Hình 2.4: Cách tính độ lệch của tổng các giá trị mỗi cột thông qua scipy

Phân phối này sẽ hoàn hảo nếu như độ lệch bằng 0, tức là model sẽ không bị thiên vị trong quá trình học khiến nó chỉ dự đoán ra 3 bounding box.

Thông thường, các Nhà phân tích Dữ liệu sẽ có một kỹ thuật Log Scale trực y để tạo ra biểu đồ không bị lệch quá nhiều:

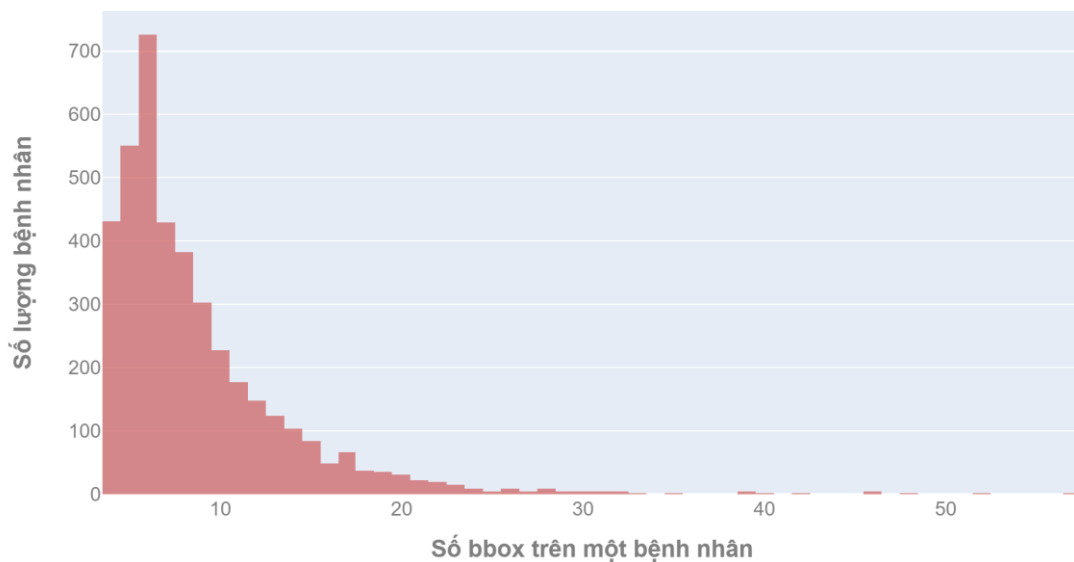
PHÂN PHỐI CỦA SỐ LƯỢNG BBOX TRÊN MỖI BỆNH NHÂN (Có Log Scale trực y)



Hình 2.5: Phân phối đang xét những có thực hiện log scale trực y

Tuy nhiên kỹ thuật này sẽ đem đến một cái nhìn không chân thật. Cách xử lý thứ hai trong trường hợp này đó là loại bỏ trường hợp $x=3$:

PHÂN PHỐI CỦA SỐ LƯỢNG BBOX TRÊN MỖI BỆNH NHÂN (Bỏ giá trị 3 và không Log Scale trực y)



Hình 2.6: Biểu đồ mới sau khi bỏ $x=3$.

Có thể thấy, số các bệnh nhân có 6 bounding box sẽ nhiều thứ 2, từ đó giảm dần số lượng nếu xét số bounding box cao hơn.

2.2. ĐẾM SỐ CĂN BỆNH TRÊN 1 BỨC ẢNH

Hãy tìm phân phối của **số lượng căn bệnh trên một bệnh nhân**. Điều này là cần thiết bởi vì nếu chỉ biết số lượng bounding box trên mỗi bệnh nhân thì sẽ không thể hiện bệnh nhân đó có bao nhiêu bệnh. Ví dụ, nếu một bác sĩ xác định bệnh nhân này có 8 khối u nhỏ nằm rải rác trên phổi, chúng ta cần đếm đó là 1 căn bệnh.

Thực hiện viết câu truy vấn liệt kê hết các căn bệnh riêng rẽ trên mỗi UID, tức là trên mỗi bên nhân:

```
[27]: # Viết câu truy vấn liệt kê hết các căn bệnh riêng rẽ trên mỗi UID, tức là trên mỗi bên nhân.
train_df.groupby('image_id')['class_id'].unique()

[27]: image_id
000434271f63a053c4128a0ba6352c7f      [14]
00053190460d56c53cc3e57321387478      [14]
0005e8e3701dfb1dd93d53e2ff537b6e      [7, 8, 6, 4]
0006e0a85696f6bb578e84fafa9a5607      [14]
0007d316f756b3fa0baea2ff514ce945      [13, 11, 3, 0, 5]
...
ffe6f9fe648a7ec29a50feb92d6c15a4      [3, 0, 9]
ffea246f04196af602c7dc123e5e48fc      [14]
ffeffc54594deb3716d6fcd2402a99f      [0]
fff0f82159f9083f3dd1f8967fc54f6a      [14]
fff2025e3c1d6970a8a6ee0404ac6940      [14]
Name: class_id, Length: 15000, dtype: object
```

Hình 2.7: Đếm số căn bệnh trên mỗi bệnh nhân.

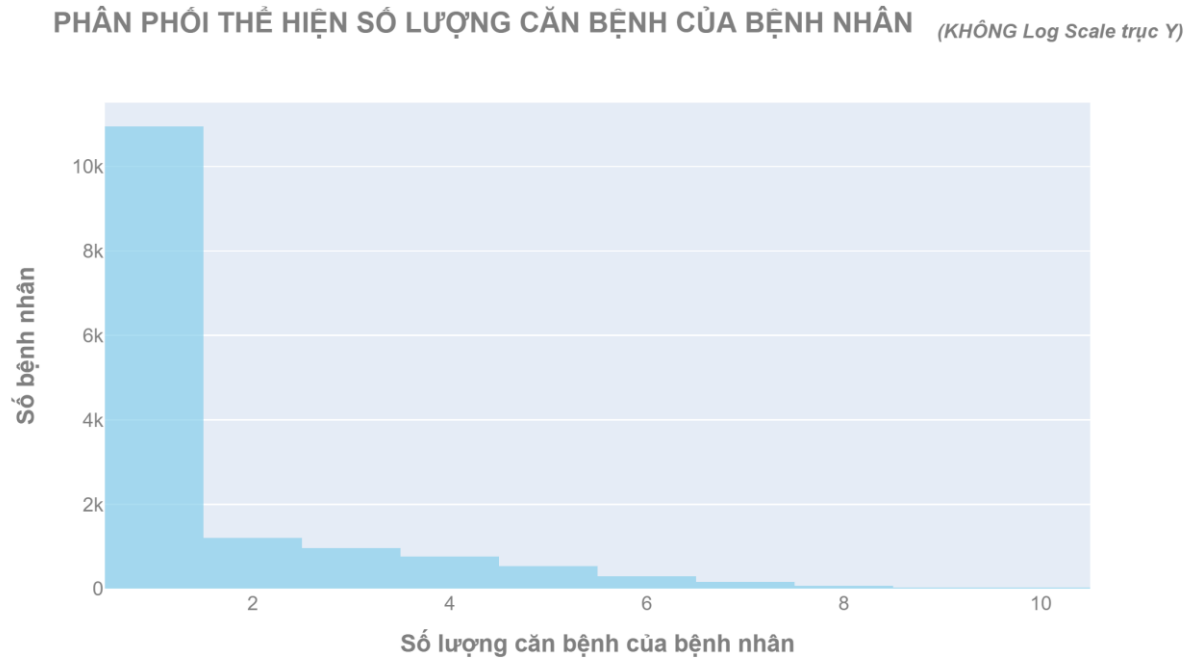
Tuy nhiên chúng ta cần cột thứ hai là một số chứ không phải một danh sách. Do đó chúng ta thực hiện lấy length của danh sách này.

```
[28]: train_df.groupby('image_id')['class_id'].unique().apply(lambda x: len(x))

[28]: image_id
000434271f63a053c4128a0ba6352c7f      1
00053190460d56c53cc3e57321387478      1
0005e8e3701dfb1dd93d53e2ff537b6e      4
0006e0a85696f6bb578e84fafa9a5607      1
0007d316f756b3fa0baea2ff514ce945      5
...
ffe6f9fe648a7ec29a50feb92d6c15a4      3
ffea246f04196af602c7dc123e5e48fc      1
ffeffc54594deb3716d6fcd2402a99f      1
fff0f82159f9083f3dd1f8967fc54f6a      1
fff2025e3c1d6970a8a6ee0404ac6940      1
Name: class_id, Length: 15000, dtype: int64
```

Hình 2.8: Đếm số căn bệnh trên mỗi bệnh nhân

Kết quả là đồ thị sau:



Hình 2.9: Đồ thị thể hiện phân phối của số lượng căn bệnh trên mỗi bệnh nhân

Tuy nhiên, đây là một phân phối không chính xác.

Nguyên nhân đến từ việc class 14 cũng được tính là một record, nếu ta không cân nhắc trường hợp này thì cột đầu tiên sẽ thể hiện cả thấy số lượng bệnh nhân có 1 căn bệnh duy nhất cộng với số lượng bệnh nhân không có bệnh nào.

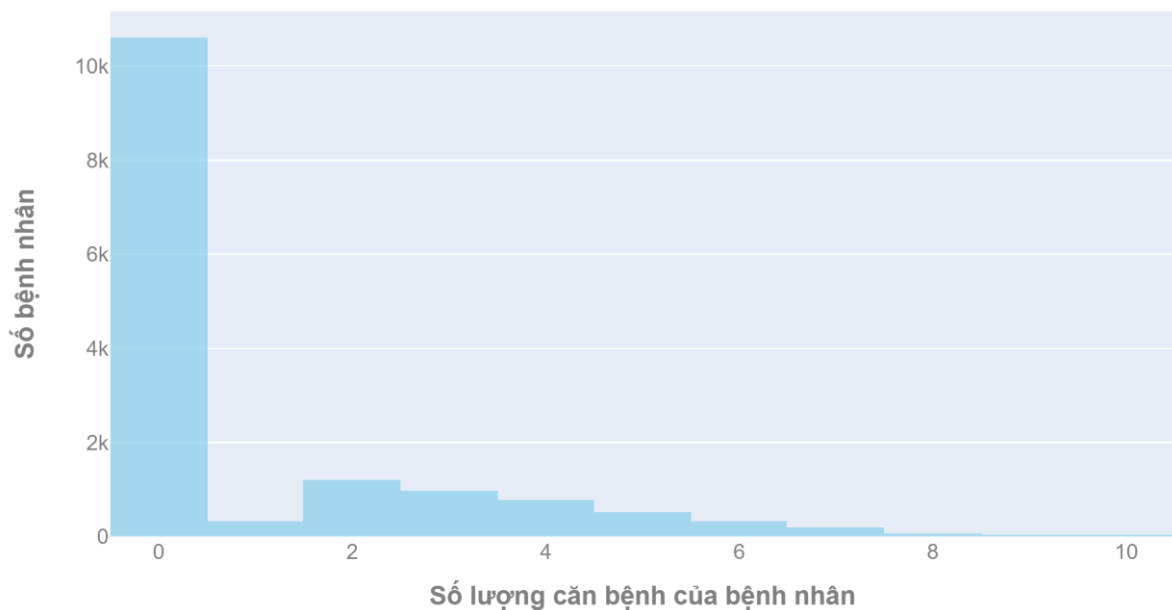
Do đó, câu truy vấn mới sẽ đem đến biểu đồ chính xác hơn như sau:

```
[64]: train_df.groupby('image_id')['class_id'].unique()\
      .apply(lambda x: len(list(filter(lambda a: a != 14, x))))
```

```
[64]: image_id
000434271f63a053c4128a0ba6352c7f    0
00053190460d56c53cc3e57321387478    0
0005e8e3701dfb1dd93d53e2ff537b6e    4
0006e0a85696f6bb578e84fafa9a5607    0
0007d316f756b3fa0baea2ff514ce945    5
..
ffe6f9fe648a7ec29a50feb92d6c15a4    3
ffea246f04196af602c7dc123e5e48fc    0
ffeffc54594debf3716d6fcd2402a99f    1
fff0f82159f9083f3dd1f8967fc54f6a    0
fff2025e3c1d6970a8a6ee0404ac6940    0
Name: class_id, Length: 15000, dtype: int64
```

Hình 2.10: Câu truy vấn mới

PHÂN PHỐI CỦA SỐ LƯỢNG CĂN BỆNH CỦA BỆNH NHÂN (KHÔNG Log Scale trục Y)



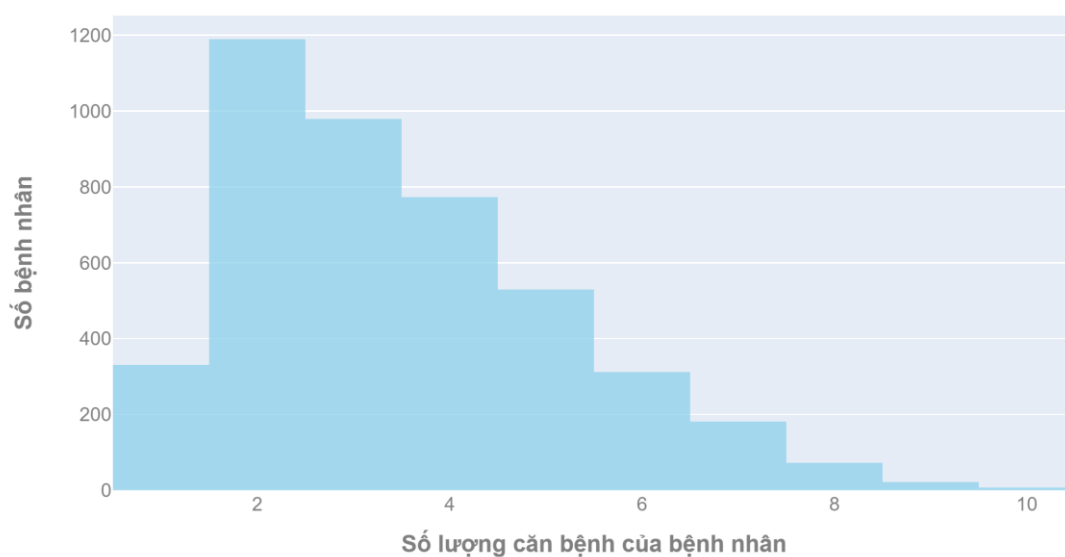
Hình 2.11: Đồ thị thể hiện phân phối của số lượng căn bệnh trên mỗi bệnh nhân

Từ biểu đồ trên, có thể rút ra được những thông tin sau:

- Số lượng bệnh nhân không có bệnh gì chiếm đa số trong tập train.
- Cũng trong tập train, một bệnh nhân có không quá 10 căn bệnh cùng lúc.
- Nếu không xét đến trường hợp no finding, số lượng bệnh nhân mắc 2 bệnh là nhiều nhất.

Biểu đồ dưới đây dùng để quan sát kĩ hơn phân phối khi không xét no_finding:

PHÂN PHỐI CỦA SỐ LƯỢNG CĂN BỆNH CỦA BỆNH NHÂN (LOẠI BỎ CLASS 14, KHÔNG Log Scale trục Y)



Hình 2.12: Đồ thị thể hiện phân phối của số lượng căn bệnh trên mỗi bệnh nhân không xét no finding

Chương 3: THĂM DÒ CỘT CLASS_NAME

Cột class_name tượng trưng cho label ở dạng string tương ứng với bounding box. Ở cột này, sinh viên sẽ thống kê số bounding box trên mỗi class

[32]:

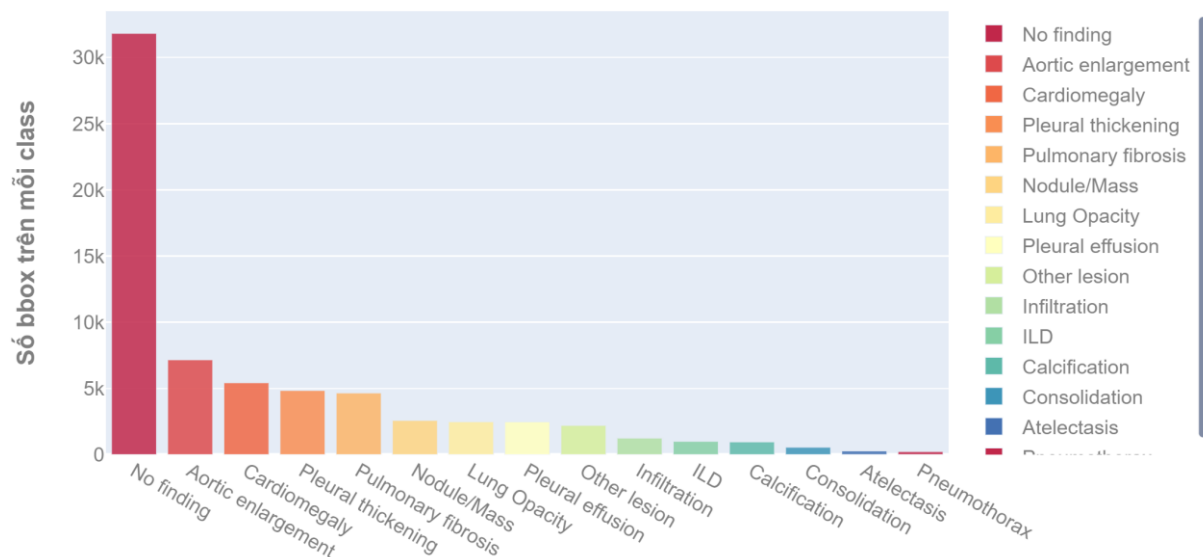
```
# Câu truy vấn như sau:
train_df.class_name.value_counts()
```

```
[32]: No finding          31818
      Aortic enlargement  7162
      Cardiomegaly       5427
      Pleural thickening  4842
      Pulmonary fibrosis  4655
      Nodule/Mass        2580
      Lung Opacity       2483
      Pleural effusion    2476
      Other lesion       2203
      Infiltration       1247
      ILD                1000
      Calcification      960
      Consolidation      556
      Atelectasis        279
      Pneumothorax       226
      Name: class_name, dtype: int64
```

Hình 3.1: Câu truy vấn số lượng bounding box trên mỗi class

Sau đó là trực quan hóa lê biểu đồ:

PHÂN PHỐI SỐ LƯỢNG BBOX THEO CLASS



Hình 3.2: Biểu đồ thể hiện phân phối số lượng bounding box theo mỗi class

Từ biểu đồ chúng ta có thể quan sát được một sự mất cân bằng giữa các class rất nghiêm trọng. Theo đó, ngoài việc No finding chiếm đa số, bệnh phình động mạch chủ sẽ chiếm số lượng nhiều nhất với hơn 7k bounding box. Trong khi đó, bệnh tràn khí phổi chiếm số lượng ít nhất với vỏn vẹn 226 bounding box.

Chương 4: THĂM DÒ CỘT CLASS_ID

Cột class_id cho biết label của bounding box được mã hóa thành số, vì thế chúng ta sẽ gỡ bỏ cột class_name. Trước khi loại bỏ, sinh viên sẽ tạo các dict để mapping từ số về lại chuỗi nếu cần thiết.

```
... [int_2_str]...           ... [str_2_int]...           ... [str_2_clr]...

{0: 'Aortic enlargement',   {'Aortic enlargement': 0,   {0: 'rgb(193, 39, 74)',
 1: 'Atelectasis',          'Atelectasis': 1,          1: 'rgb(221, 74, 76)',
 2: 'Calcification',        'Calcification': 2,        2: 'rgb(240, 103, 68)',
 3: 'Cardiomegaly',         'Cardiomegaly': 3,        3: 'rgb(249, 142, 82)',
 4: 'Consolidation',        'Consolidation': 4,        4: 'rgb(253, 181, 103)',
 5: 'ILD',                  'ILD': 5,                  5: 'rgb(254, 212, 129)',
 6: 'Infiltration',         'Infiltration': 6,        6: 'rgb(254, 236, 159)',
 7: 'Lung Opacity',         'Lung Opacity': 7,        7: 'rgb(255, 255, 190)',
 8: 'Nodule/Mass',          'Nodule/Mass': 8,        14: 'rgb(239, 249, 166)',
 9: 'Other lesion',         'Other lesion': 9,        8: 'rgb(214, 238, 155)',
10: 'Pleural effusion',     'Pleural effusion': 10,   9: 'rgb(177, 223, 163)',
11: 'Pleural thickening',   'Pleural thickening': 11, 10: 'rgb(134, 207, 165)',
12: 'Pneumothorax',        'Pneumothorax': 12,      11: 'rgb(94, 185, 169)',
13: 'Pulmonary fibrosis',   'Pulmonary fibrosis': 13, 12: 'rgb(61, 149, 184)',
14: 'No finding'}          'No finding': 14}        13: 'rgb(68, 113, 178)'}
```

Hình 4.1: Các dictionary cần thiết được tạo ra trước khi loại bỏ cột class_name

	image_id	class_id	rad_id	x_min	y_min	x_max	y_max
0	50a418190bc3fb1ef1633bf9678929b3	14	R11	NaN	NaN	NaN	NaN
1	21a10246a5ec7af151081d0cd6d65dc9	14	R7	NaN	NaN	NaN	NaN
2	9a5094b2563a1ef3ff50dc5c7ff71345	3	R10	691.0	1375.0	1653.0	1831.0
3	051132a778e61a86eb147c7c6f564dfe	0	R10	1264.0	743.0	1611.0	1019.0
4	063319de25ce7edb9b1c6b8881290140	14	R10	NaN	NaN	NaN	NaN

Hình 4.2: Head của Train Dataframe sau khi drop cột class_name

Chương 5: THĂM DÒ CỘT CLASS_ID

Cột `rad_id` tượng trưng cho ID của bác sĩ x-quang. Các giá trị này được mã hóa từ R1 đến R17 tượng trưng cho 17 bác sĩ x-quang thực hiện gắn nhãn. Ở mục này, chúng tôi sẽ thực hiện tìm phân phối của số lượng annotation được đánh bởi các bác sĩ. Ngoài ra, chúng tôi sẽ tích hợp thêm thông tin số lượng mỗi class mà bác sĩ đó đã gắn nhãn.

5.1. SỐ LƯỢNG ANNOTATION TRÊN MỖI BÁC SĨ

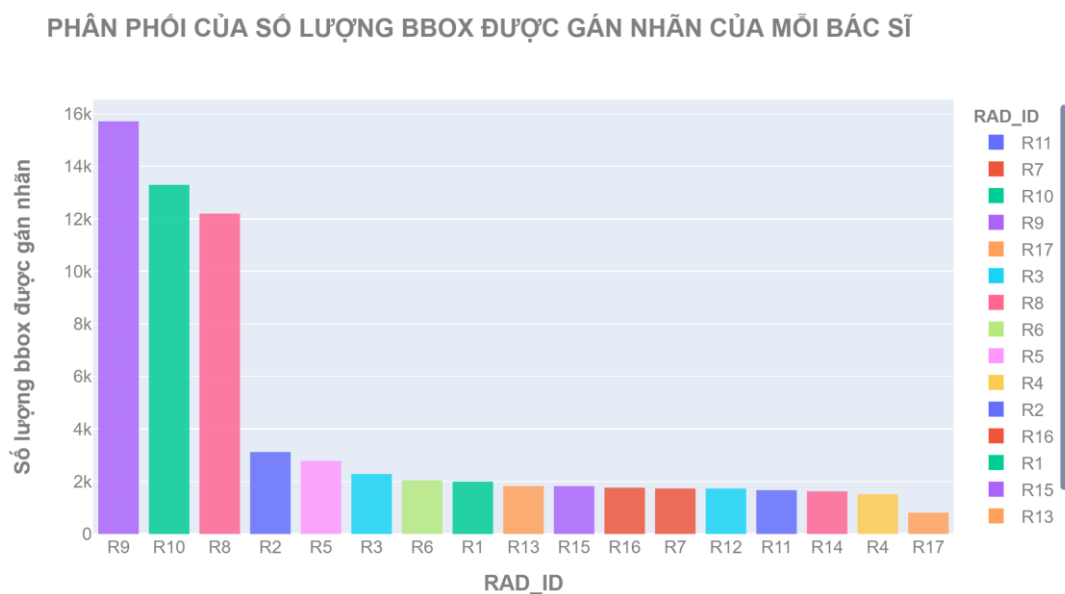
Đầu tiên, thống kê cột `rad_id`:

```
[35]: train_df["rad_id"].value_counts()
```

```
[35]: R9      15708
      R10     13292
      R8      12198
      R2       3121
      R5       2783
      R3       2285
      R6       2041
      R1       1995
      R13      1824
      R15      1823
      R16      1763
      R7       1733
      R12      1729
      R11      1670
      R14      1624
      R4       1513
      R17        812
      Name: rad_id, dtype: int64
```

Hình 5.1: Kết quả của câu truy vấn

Sau đó vẽ biểu đồ:



Hình 5.2: Biểu đồ thể hiện phân phối của số lượng bounding box gom theo `rad_id`.

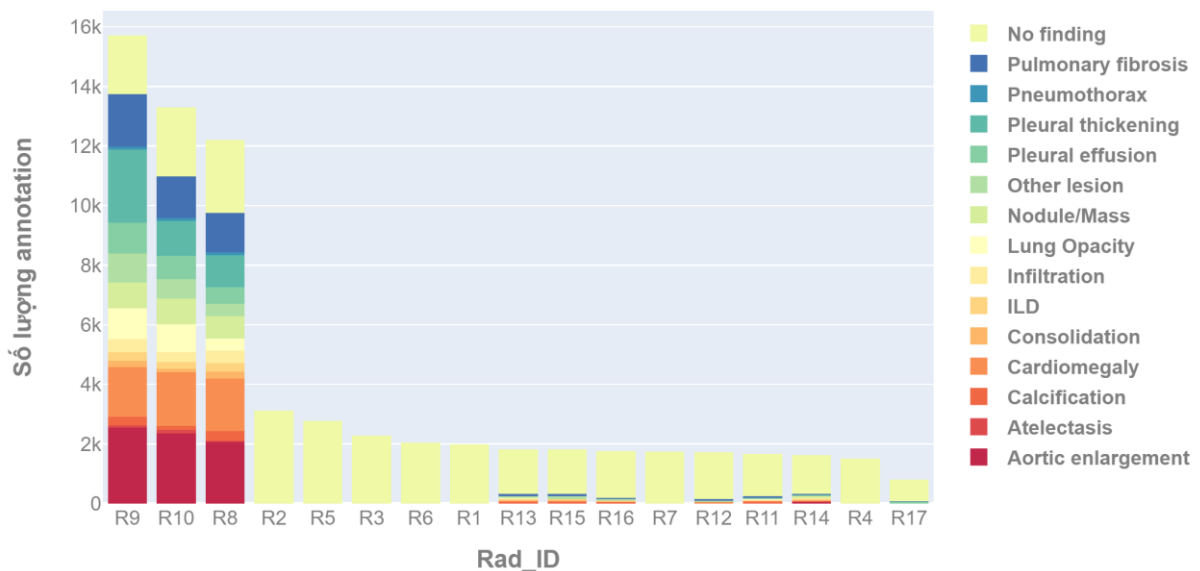
Từ biểu đồ trên, có thể rút ra được những thông tin sau:

- 3 bác sĩ x-quang (R9, R10, & R8 theo thứ tự) đảm nhận khối lượng công việc gán nhãn lớn nhất (~40-50% tổng số annotation)
- Các bác sĩ còn lại đảm nhận công việc tương đồng nhau

Tuy nhiên, biểu đồ này vẫn chưa thể hiện phân phối các class mà mỗi bác sĩ gán nhãn. Do đó hãy cùng xem xét thông tin này.

5.2. SỐ LƯỢNG ANNOTATION TRÊN MỖI BÁC SĨ (CÓ THÔNG TIN VỀ CLASS)

PHÂN PHỐI CỦA SỐ LƯỢNG ANNOTATION CÙNG CLASS_ID BỞI MỖI BÁC SĨ




Hình 5.3: Biểu đồ thể hiện phân phối của số lượng bounding box gom theo rad_id có thể hiện class_id.

Từ biểu đồ trên, có thể rút ra được những thông tin sau:

- 3 bác sĩ x-quang (R9, R10, & R8 theo thứ tự) không chỉ đảm nhận khối lượng công việc gán nhãn lớn nhất mà còn bao phủ hết tất cả 14 loại bệnh, với tỷ lệ gần như nhau.
- Các bác sĩ còn lại đảm nhận việc gán nhãn cho các bức ảnh no finding là chính.

Chương 6: THĂM DÒ CÁC CỘT TỌA ĐỘ BOUNDING BOX

Ở bước này, sinh viên sẽ thêm vào `train_df` 2 cột `img_height` và `img_width`, đồng thời thêm vào 4 cột nữa thể hiện các giá trị `top`, `left`, `bottom`, `right` ở dạng thập phân.

100%  4394/4394 [09:18<00:00, 11.30it/s]

Đang xử lý 2 cột `width` và `height`
Đang xử lý 4 cột `bbox` tính theo tỷ lệ phần trăm

[39]:

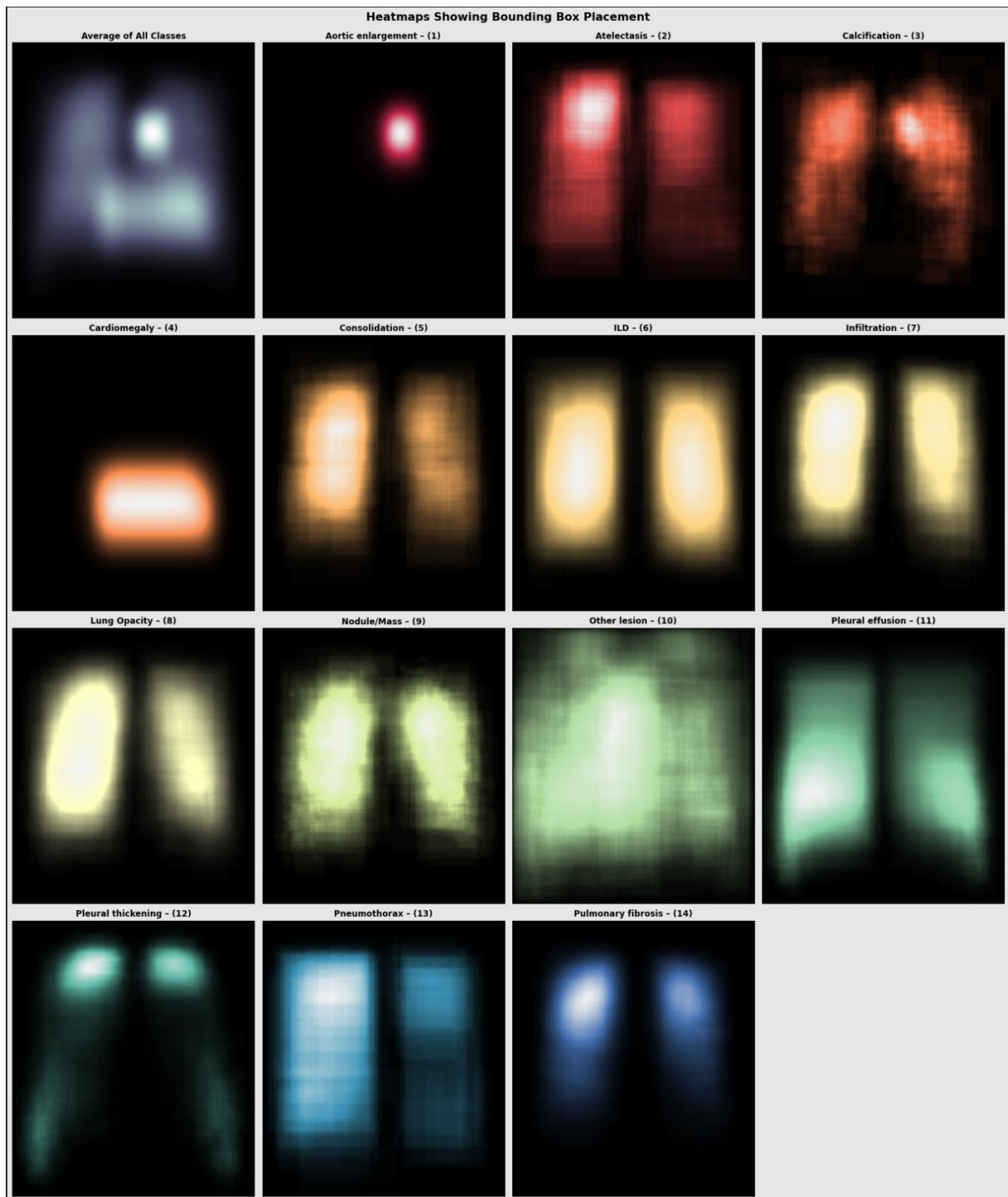
	<code>image_id</code>	<code>class_id</code>	<code>rad_id</code>	<code>x_min</code>	<code>y_min</code>	<code>x_max</code>	<code>y_max</code>	<code>img_height</code>	<code>img_width</code>	<code>frac_x_min</code>	<code>frac_x_max</code>	<code>frac_y_min</code>	<code>frac_y_max</code>
	2563a1ef3ff50dc5c7ff71345	3	R10	691.0	1375.0	1653.0	1831.0	2336	2080	0.332212	0.794712	0.588613	0.783818
	8e61a86eb147c7c6f564dfe	0	R10	1264.0	743.0	1611.0	1019.0	2880	2304	0.548611	0.699219	0.257986	0.353819
	af4ce1a3030eb8167753b06	11	R9	627.0	357.0	947.0	433.0	3072	2540	0.246850	0.372835	0.116211	0.140951
	3d5f5e4846aa4ca6db4daf1	5	R17	1347.0	245.0	2188.0	2169.0	2555	2285	0.589497	0.957549	0.095890	0.848924
	1cbeec15182ed335a8b5a9e	8	R9	557.0	2352.0	675.0	2484.0	3353	2568	0.216900	0.262850	0.701461	0.740829

Hình 6.1: Data frame mới

Ngoài ra, sinh viên đã phân tích các thông tin liên quan đến cột này ở chương 5.

Chương 7: CÁC TRỰC QUAN HÓA DỮ LIỆU KHÁC

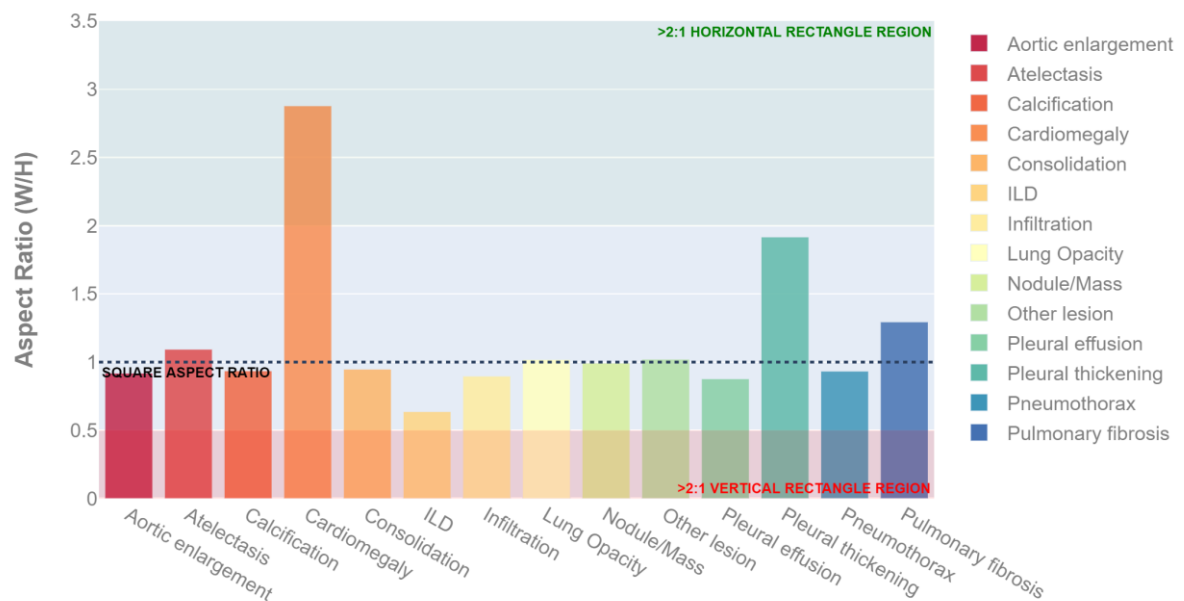
Để hình dung rõ hơn về vị trí và tần xuất xuất hiện của mỗi class, một biểu đồ headmap sẽ được lập như sau:



Hình 7.1: Headmap thể hiện vị trí và mật độ của các bounding box phân theo từng căn bệnh.

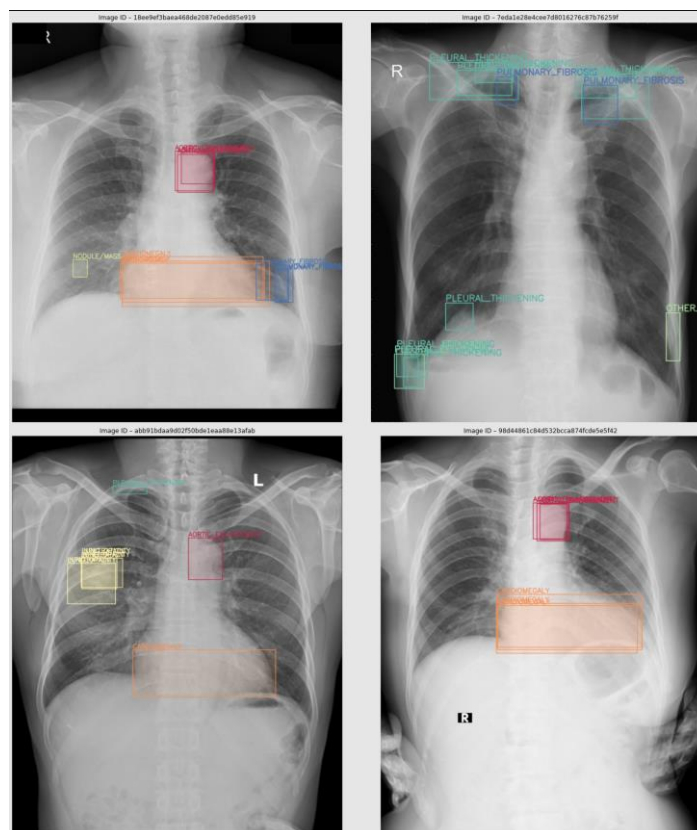
Ngoài ra, để có cái nhìn về tỷ lệ của các bounding box thuộc mỗi class, một biểu đồ cột thể hiện phân phối của các tỷ lệ bounding box như sau:

phân phối của tỷ lệ lên bbox phân theo class

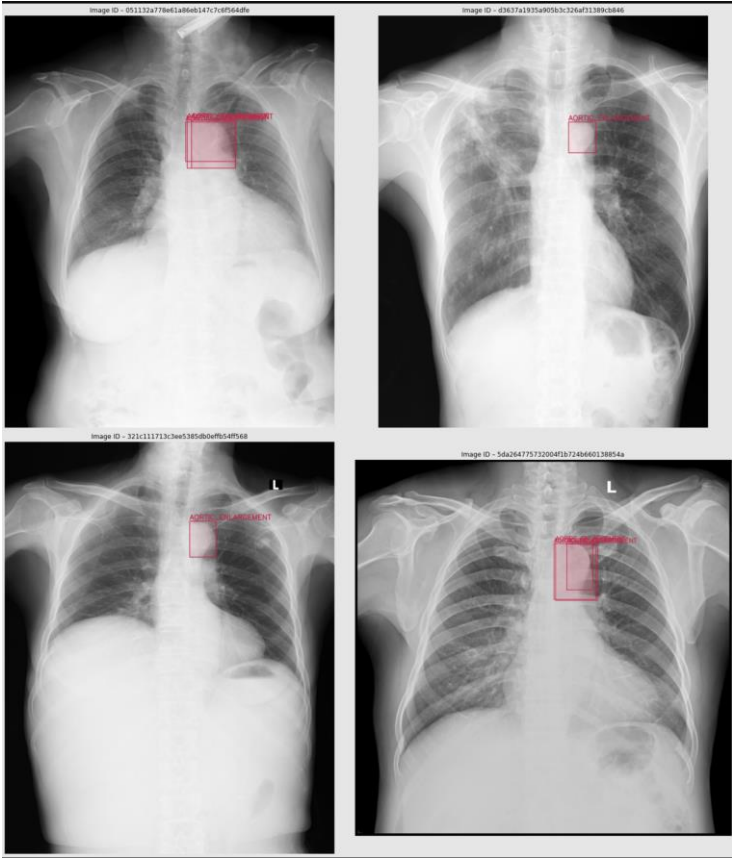


Hình 7.2: Biểu đồ thể hiện phân phối của tỷ lệ lên bounding box phân theo class

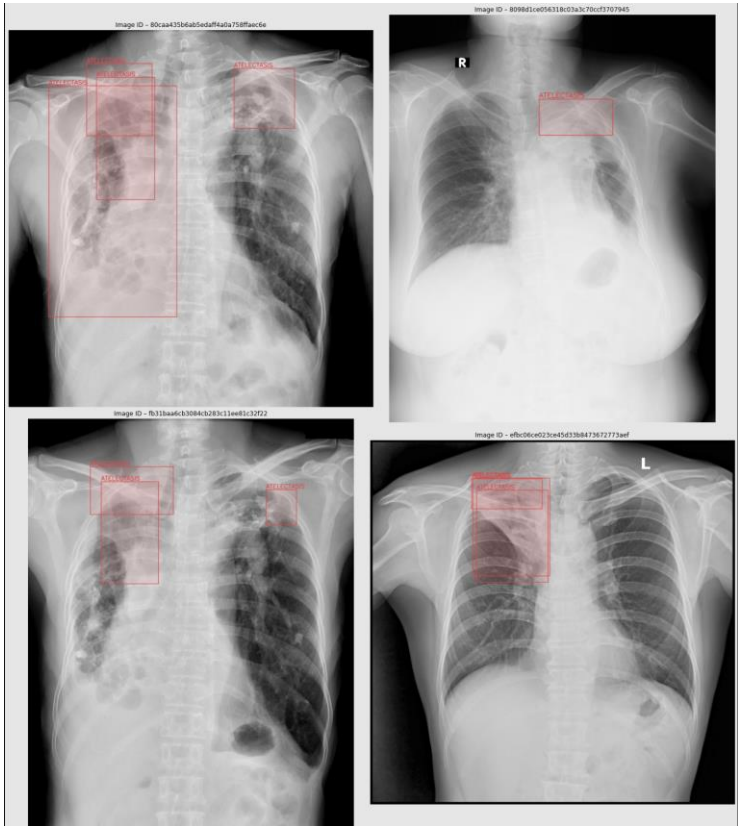
Sau cùng, chúng ta hãy cùng vẽ bounding box lên một số ảnh để quan sát:



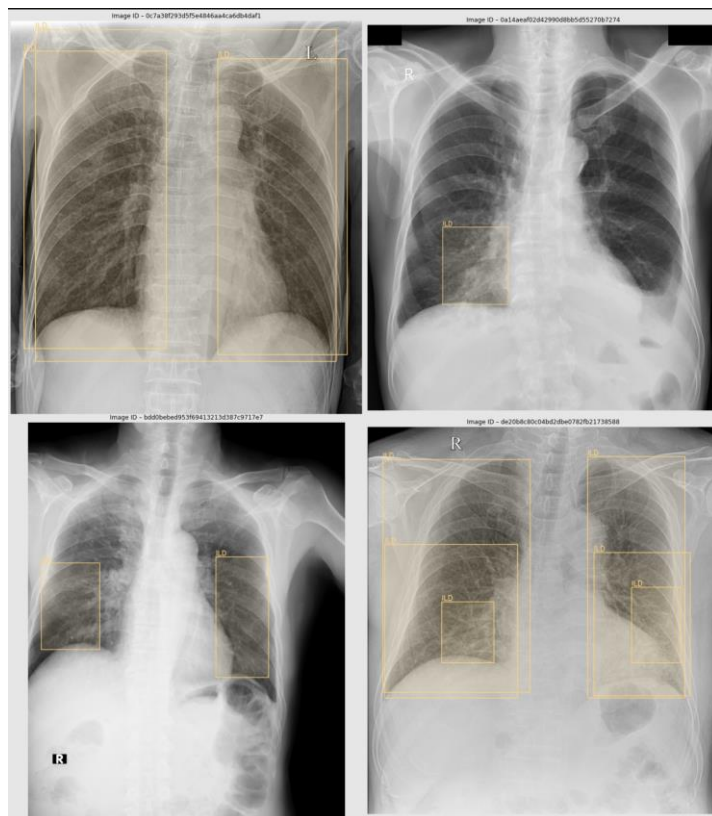
Hình 7.3: Ảnh được lấy ngẫu nhiên



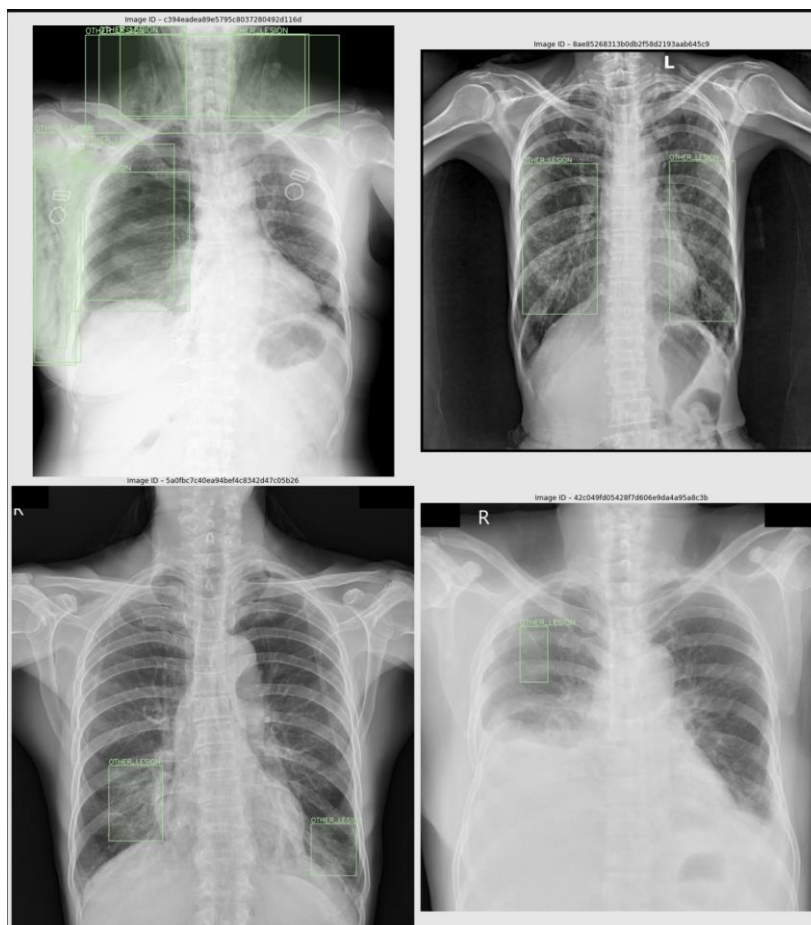
Hình 7.4: Các ảnh thuộc class



Hình 7.5: Các ảnh thuộc class 1



Hình 7.6: Các ảnh thuộc class 5



Hình 7.7: Các ảnh thuộc class 9

TÀI LIỆU THAM KHẢO

- [1] Trang chủ cuộc thi: <https://www.kaggle.com/c/vinbigdata-chest-xray-abnormalities-detection>
- [2] Pandas documentation: <https://pandas.pydata.org/docs/>
- [3] Plotly documentation: <https://plotly.com/python/>
- [4] <https://www.kaggle.com/trungthanhnguyen0502/eda-vinbigdata-chest-x-ray-abnormalities>
- [5] Chest_X-ray_Starter: <https://www.kaggle.com/kostiantynperun/chest-x-ray-starter>