

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA TOÁN – CƠ – TIN HỌC**



**BÁO CÁO MÔN HỌC**

**ĐỀ TÀI: DỰ ĐOÁN KHẢ NĂNG ĐƯỢC NHẬN NUÔI CỦA ĐỘNG  
VẬT TỪ TRẠM CỨU HỘ VỚI DỮ LIỆU ĐẦU VÀO ĐA DẠNG**

*Học viên:*

Phạm Quang Hiếu

Hà Nội - 2022

**MỤC LỤC**

GIỚI THIỆU VÀ MỤC TIÊU BÀI TOÁN.....	2
1. TỔNG QUAN.....	3
1.1. Phát biểu bài toán.....	3
1.2. Giới thiệu về bộ dữ liệu .....	3
1.2.1. Dữ liệu bảng .....	3
1.2.2. Hình ảnh .....	5
1.2.3. Metadata ảnh .....	5
1.2.4. Dữ liệu mô tả .....	5
2. THƯỚC ĐO .....	6
2.1. Các thước đo phổ biến cho bài toán phân loại.....	6
2.1.1. Accuracy.....	6
2.1.2. Confusion matrix .....	6
2.1.3. TPR, FPR, TNR, FNR.....	7
2.1.4. ROC, AUC .....	8
2.1.5. Precesion và Recall .....	8
2.2. Quadratic weighted kappa .....	9
3. PHƯƠNG PHÁP XỬ LÝ.....	10
3.1. Sử dụng Functional API trong mạng học sâu.....	10
3.2. Trích chọn đặc trưng.....	11
4. DỮ LIỆU METADATA ẢNH.....	12
4.1. faceAnnotations .....	12
4.2. LabelAnnotations.....	13
4.3. TextAnnotations.....	14
4.4. ImagePropertiesAnnotation .....	16
4.5. cropHintsAnnotation.....	17
5. DỮ LIỆU SENTIMENT .....	18
5.1. Phân tích cảm xúc .....	18
5.2. Phân tích thực thể .....	19
6. PHÂN TÍCH DỮ LIỆU DẠNG BẢNG.....	21
6.1. Khai phá và xử lý dữ liệu missing .....	21
6.1.1. Tổng quan về bộ dữ liệu.....	21

1.2.2. Kiểm tra và xử lý các giá trị null (rỗng) trong bộ dữ liệu .....	22
6.2. Phân tích biến đơn và xử lý dữ liệu .....	23
6.2.1. Biến Type .....	23
6.2.2. Biến Age .....	24
6.2.3. Biến giống loài (Breed) .....	25
6.2.4. Biến Gender.....	25
6.2.5. Biến màu lông (color1, color2, color3) .....	27
6.2.6. Biến độ dài lông (FurLength).....	28
6.2.7 Các biến liên quan tới sức khỏe vật nuôi (Vaccinated, Dewormed, Health) .	28
6.2.8. Số lượng con vật (Quantity) .....	29
6.2.9. Biến phí (Fee):.....	30
6.2.10. Vùng – địa điểm nhận nuôi (State).....	31
6.2.11. Các thông số về hình ảnh của con vật (VideoAmt, PhotoAmt) .....	31
6.2.12. Biến Name .....	32
6.2.13. Biến tốc độ nhận nuôi (AdoptionSpeed) .....	33
6.3. Trích chọn đặc trưng cho mô hình (Feature engineering) .....	33
6.3.1. Các đặc trưng đã qua xử lý:.....	33
6.3.2. Phân tích tương quan giữa các biến giải thích với biến phụ thuộc: .....	35
Hình 6.22. Tương quan biến AdoptionSpeed với các biến còn lại.....	35
7. PHÂN TÍCH TRÍCH CHỌN ĐẶC TRƯNG TỪ DỮ LIỆU ẢNH.....	37
7.1. Giới thiệu bộ dữ liệu: .....	37
7.2. Tổng quan về deep learning – mạng tích chập CNN.....	37
7.3. Khó khăn thách thức .....	41
7.4. Xử lý dữ liệu đầu vào.....	42
7.4.1. Chuẩn hóa kích thước hình ảnh:.....	42
7.4.2. Chuẩn hóa kích thước hình ảnh:.....	42
7.5. Xây dựng mô hình .....	45
8. PHÂN TÍCH DỮ LIỆU DẠNG VĂN BẢN .....	47
8.1. Tổng quan về xử lý ngôn ngữ tự nhiên (NLP) .....	47
8.1.1. N-gram.....	47
8.1.2. Trích rút từ khóa trong văn bản.....	48

8.1.3. Tiền xử lý văn bản và Nhận dạng cụm từ Ứng viên .....	49
8.1.4. Nhận dạng cụm từ Ứng viên .....	49
8.1.5. Các đặc điểm cụm từ phổ biến .....	49
8.1.6. TF-IDF .....	50
8.2. Phân tích trường dữ liệu Description .....	51
8.2.1. Những từ phổ biến nhất .....	51
8.2.1. Tương quan giữa độ dài của mô tả với tốc độ con vật được nhận nuôi .....	53
8.2.3. Sinh mô hình TF-IDF từ trường Description .....	53
9. XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH .....	54
9.1. Các mô hình sử dụng .....	54
9.1.1. Support vector machine (SVM) .....	54
9.1.2. Random Forest .....	54
9.1.3. LightGBM .....	55
9.1.4. Kết hợp mô hình (ensemble) .....	56
9.2. Phương thức đánh giá .....	57
9.2.1. Phân tách dữ liệu (train-test split) .....	57
9.2.2. Kiểm chứng chéo (Cross validation) .....	58
9.3. Kết quả đánh giá .....	59
10. KẾT LUẬN .....	60
TÀI LIỆU THAM KHẢO .....	61

## DANH MỤC HÌNH ẢNH

Hình 2.1. Thước đo độ chính xác .....	6
Hình 2.2. Confusion matrix trong bài toán phân loại .....	7
Hình 2.3. Chỉ số TPR, FPR trong phân loại 2 lớp .....	7
Hình 2.4. Đồ thị chỉ số ROC, AUC .....	8
Hình 2.5. Cách tính precision và recall .....	9
Hình 2.6. Ma trận trọng số của chỉ số Kappa trong bài .....	10
Hình 3.1. Ví dụ mô hình sử dụng Sequential API .....	10
Hình 3.2. Mô hình 2 kiểu dữ liệu đầu vào sử dụng Functional API .....	11
Hình 3.3. Xử lý mỗi nguồn dữ liệu bằng một mô hình riêng trước khi kết hợp .....	12
Hình 4.1. Đặc trưng faceAnnotations .....	12
Hình 4.2. Thời gian chờ nhận nuôi metadata faceAnnotation .....	13
Hình 4.3. Đặc trưng labelAnnotations .....	13
Hình 4.4. Thời gian chờ nhận nuôi theo nhãn metadata .....	14
Hình 4.5. TEXT_DETECTION OCR .....	14
Hình 4.6. DOCUMENT_TEXT_DETECTION OCR .....	15
Hình 4.7. Thời gian chờ nhận nuôi theo metadata textAnnotation .....	15
Hình 4.8. Đặc trưng imagePropertiesAnnotation .....	16
Hình 4.9. Số lượng ảnh theo màu chủ đạo .....	16
Hình 4.10. Thời gian chờ nhận nuôi theo màu chủ đạo của ảnh .....	17
Hình 4.11. Đặc trưng cropHintsAnnotation .....	17
Hình 5.1. Biểu diễn giải mức độ cảm xúc theo các giá trị mẫu .....	19
Hình 5.2. Phân bổ các nhóm theo thực thể được xác định trong mô tả .....	20
Hình 5.3. Thời gian chờ theo ngôn ngữ trong phần mô tả .....	21
Hình 6.1. Kiểm tra các quan sát đầu và cuối trong bộ dữ liệu .....	21
Hình 6.2. Kiểm tra các giá trị null (rỗng) trong bộ dữ liệu .....	22
Hình 6.3. Biểu đồ biểu diễn số lượng chó, mèo trong bộ dữ liệu .....	23
Hình 6.4. Biểu đồ tuổi của vật nuôi theo tháng .....	24
Hình 6.5. Biểu đồ tuổi của vật nuôi theo năm .....	24
Hình 6.6. Biểu đồ biểu diễn phân phối của 2 biến về giống loài .....	25
Hình 6.7. Biểu đồ biểu diễn tỉ lệ giới tính của vật nuôi .....	26
Hình 6.8. Biểu đồ biểu diễn tỉ lệ giới tính của vật nuôi theo thời gian được nhận nuôi .....	26
Hình 6.9. Biểu đồ biểu diễn các loại màu lông .....	27
Hình 6.10. Biểu đồ biểu diễn số màu lông theo thời gian được nhận nuôi .....	27

Hình 6.11. Biểu đồ biểu diễn độ dài lông.....	28
Hình 6.13. Biểu đồ tình trạng y tế các con vật nuôi theo thời gian được nhận nuôi .....	29
Hình 6.14. Biểu đồ biểu diễn số lượng vật nuôi hiện có .....	29
Hình 6.15. Biểu đồ biểu diễn phân phối phí nhận nuôi .....	30
Hình 6.16. Biểu đồ biểu diễn phí nhận nuôi theo thời gian được nhận nuôi.....	30
Hình 6.17. Biểu đồ biểu diễn vùng nhận nuôi con vật .....	31
Hình 6.18. Biểu đồ biểu diễn thông số về hình ảnh con vật.....	31
Hình 6.19. Biểu đồ kiểu đặt tên của con vật.....	32
Hình 6.20. Biểu đồ thời gian được nhận nuôi .....	33
Hình 6.21. Biểu đồ hệ số tương quan giữa các biến định lượng .....	35
Hình 7.1. Giới thiệu bộ dữ liệu: Hình ảnh 1 số con vật được cứu hộ .....	37
Hình 7.2. Hình ảnh 1 một mảng ma trận RGB 6x6x3 .....	38
Hình 7.3. Luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị ...	38
Hình 7.4. Các lớp tích chập 1 .....	39
Hình 7.5. Các lớp tích chập 2 .....	39
Hình 7.6. Các lớp tích chập 3 .....	40
Hình 7.7. Lớp gộp.....	41
Hình 7.8. Lớp fully connected.....	41
Hình 7.9. Chuẩn hóa các kích thước ảnh.....	42
Hình 7.10. Mô hình ssd_mobilenet_v3 giúp phát hiện và khoanh vùng đối tượng chó mèo.....	44
Hình 7.11. Xây dựng mô hình .....	46
Hình 7.12. Huấn luyện tập train và tập test với mô hình đã dựng.....	46
Hình 7.13. Kết quả trên tập test .....	46
Hình 8.1. N-gram.....	47
Hình 8.2. Trích rút từ khóa trong văn bản.....	48
Hình 8.3. Những từ phổ biến nhất trong trường Description .....	52
Hình 8.4. Những từ phổ biến nhất trong trường Name .....	52
Hình 8.5. Tương quan giữa Description và Adoptionspeed .....	53
Hình 9.1 Thuật toán Support Vector Machine .....	54
Hình 9.2. Mô hình thuật toán Random Forest .....	55
Hình 9.3. Thuật toán LightGBM .....	56
Hình 9.4. Phương thức ensemble kết hợp các mô hình .....	57
Hình 9.5. Mô tả các tập Training, Validation, Testing.....	58
Hình 9.6. Đánh giá kết quả bằng cross validation.....	59

# LỜI NÓI ĐẦU

Tài liệu này là báo cáo môn học “**Học máy và khai phá dữ liệu nâng cao**” của nhóm học viên ngành Khoa học dữ liệu - Khóa 3 – Khoa Toán Cơ Tin – Trường đại học Khoa học tự nhiên.

Nội dung của báo cáo này là vận dụng các kiến thức đã học để giải quyết bài toán dự đoán thời gian các con vật tại trung tâm cứu hộ động vật được nhận nuôi bằng việc dựng mô hình dự báo dựa trên đa dạng các dữ liệu đầu vào.

Bố cục của báo cáo này gồm các phần chính sau:

1. Giới thiệu bài toán và mục tiêu của bài toán.
2. Thước đo.
3. Phương pháp xử lý.
4. Giới thiệu dữ liệu Metadata ảnh.
5. Giới thiệu dữ liệu Sentiment.
6. Phân tích các kiểu dạng dữ liệu cụ thể của bài toán.
7. Xây dựng và đánh giá mô hình.
8. Kết luận

Hi vọng tài liệu này sẽ đem đến những kiến thức hữu ích cho người đọc.

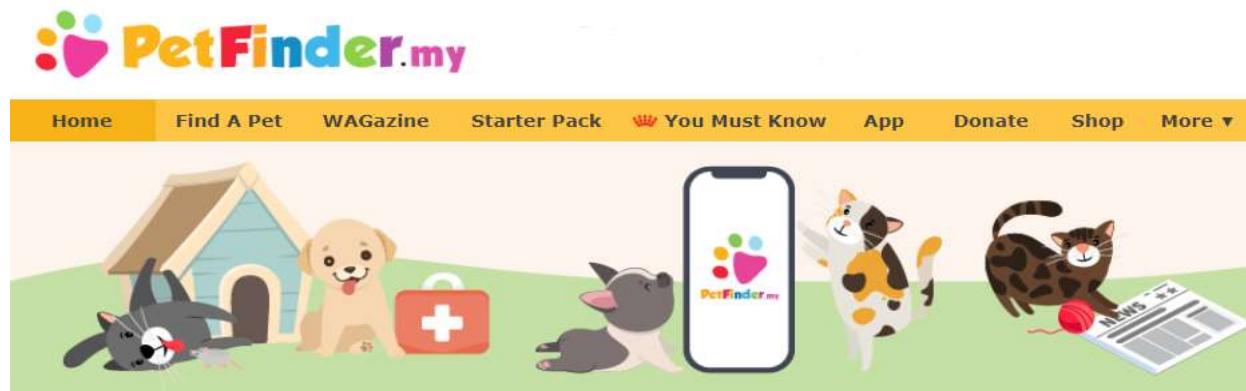
## Nhóm thực hiện

Nguyễn Thanh Tùng (Mã học viên: 20007904)

Phạm Quang Hiếu (Mã học viên: 20007919)

Lê Thị Mỹ Hạnh (Mã học viên: 20007922)

## GIỚI THIỆU VÀ MỤC TIÊU BÀI TOÁN



Theo số liệu của tổ chức WHO, trong năm 2012, toàn thế giới có khoảng 600 triệu con chó hoang. Riêng tại Mỹ, con số này 70 triệu và chỉ có khoảng 10% số động vật này được đưa vào trung tâm chăm sóc động vật mỗi năm. Như vậy, đa phần con vật sẽ sinh sống vất vưởng trên đường phố và các ngôi nhà bỏ hoang hoặc sẽ bị giết hại. Mặc dù chó, mèo là những người bạn thân thiết nhất của con người nhưng chúng cũng là hai loại con vật bị bỏ rơi nhiều nhất.

Vì tình thương yêu, nhiều nhóm cứu hộ động vật đã ra đời với nỗ lực cứu giúp và phần nào bù đắp lại những tổn thương cho các con vật được cứu hộ về dù là hoang, lạc, bị bỏ rơi hay bạo hành. Ngoài ra, các trạm cứu hộ động vật cũng luôn nỗ lực tìm những mái ấm mới cho các con vật được cứu hộ với hi vọng chúng sẽ có cơ hội được yêu thương và một tương lai tốt đẹp.

Báo cáo này được viết với mục đích cung cấp giải pháp cho việc dự đoán thời gian các con vật tại trung tâm cứu hộ động vật được nhận nuôi bằng việc dựng mô hình dự báo.

Bộ dữ liệu sử dụng để huấn luyện là thông tin về các con vật tại trang web [petfinder.my](#) được cung cấp trên Kaggle, với dữ liệu của gần 15,000 con vật và 23 biến độc lập là thành phần dữ liệu của mỗi con. Ngoài dạng bảng, bộ dữ liệu còn cung cấp hình ảnh và mô tả về con vật. Việc lựa chọn bộ dữ liệu trên Kaggle được quyết định dựa trên tính khái quát và khả năng kết quả đánh giá mô hình đã kết luận được thời gian con vật được nhận nuôi có thể dựa vào nhiều yếu tố khác nhau.

Nhóm nghiên cứu đã kết hợp các mô hình lại để dự báo thời gian dựa vào các biến thành phần trong dữ liệu và cả các biến mới được bổ sung cũng như kết hợp các kỹ thuật trong Học Máy để xây dựng và triển khai ra mô hình cuối cùng có tác dụng dự báo sát với thời gian thực tế.

## 1. TỔNG QUAN

### 1.1. Phát biểu bài toán

Hàng triệu con vật đi lạc gặp nạn trên đường phố hoặc bị giết trong những nơi trú ẩn mỗi ngày trên khắp thế giới. Nếu có thể tìm thấy nhà cho chúng, nhiều mạng sống quý giá có thể được cứu và đem lại niềm vui, hạnh phúc cho những người yêu thương động vật.

[PetFinder.my](#) là nền tảng phúc lợi động vật hàng đầu của Malaysia, kể từ năm 2008, với cơ sở dữ liệu hơn 150.000 con vật. PetFinder hợp tác chặt chẽ với những người yêu động vật, giới truyền thông, các tập đoàn và các tổ chức toàn cầu để cải thiện phúc lợi động vật.

Tỷ lệ nhận nuôi động vật có tương quan chặt chẽ với siêu dữ liệu (metadata) được liên kết với hồ sơ trực tuyến của chúng, chẳng hạn như dữ liệu mô tả và đặc điểm ảnh. PetFinder hiện đang thử nghiệm một công cụ AI đơn giản có tên là Cuteness Meter, công cụ này xếp hạng mức độ dễ thương của con vật dựa trên những phẩm chất có trong ảnh của chúng. Tuy nhiên, dự đoán thời gian chờ nhận nuôi của con vật tại trạm cứu hộ thường được dựa theo cảm tính, phương pháp này thiếu tính logic và các quy trình, tiêu chuẩn được chấp nhận do đó việc có sẵn mô hình dự báo giúp những người quản lý và chăm sóc con vật không bỏ qua những yếu tố, thông tin quan trọng giúp cải thiện khả năng con vật được nhận nuôi.

Trong báo cáo này, nhóm sẽ phát triển các thuật toán để dự đoán khả năng nhận nuôi của động vật được cứu hộ \_ cụ thể là tốc độ mà con vật được nhận nuôi \_ với mong muốn cung cấp công cụ AI cho các tổ chức cứu hộ, từ đó giúp nâng cao hiệu quả của việc tìm chủ nhân mới cho con vật.

### 1.2. Giới thiệu về bộ dữ liệu

Dữ liệu được sử dụng trong báo cáo tên được cung cấp từ tổ chức PetFinder. Dữ liệu bao gồm văn bản, bảng và hình ảnh. Cụ thể:

#### 1.2.1. Dữ liệu bảng

TÊN TRƯỜNG	MÔ TẢ
PetID	ID của mỗi con vật.
AdoptionSpeed	Tốc độ nhận nuôi là trường thế hiện con vật có được nhận nuôi hay không và tốc độ nhận nuôi như thế nào. Tốc độ nhận nuôi được phân loại, chỉ số thấp hơn cho thấy tốc độ nhận nuôi nhanh hơn. <b>Đây là giá trị để dự đoán.</b>
Type	Loại động vật (1 = Chó, 2 = Mèo).

*Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng*

TÊN TRƯỜNG	MÔ TẢ
Name	Tên con vật (Bỏ trống nếu không có tên).
Age	Tuổi của con vật khi được liệt kê, tính bằng tháng.
Breed1	Giống loại chính của con vật.
Breed2	Giống loại thứ cấp của con vật nếu nó thuộc giống hỗn hợp.
Gender	Giới tính của con vật (1 = Đực, 2 = Cái, 3 = Hỗn hợp_nếu hồ sơ là đại diện cho nhóm của các con vật).
Color1	Màu lông 1 của con vật.
Color2	Màu lông 2 của con vật.
Color3	Màu lông 3 của con vật.
MaturitySize	Kích thước của con vật khi trưởng thành (1 = Nhỏ, 2 = Trung bình, 3 = Lớn, 4 = Rất lớn, 0 = Không rõ).
FurLength	Chiều dài lông (1 = Ngắn, 2 = Trung bình, 3 = Dài, 0 = Không rõ).
Vaccinated	Con vật đã được tiêm phòng (1 = Có, 2 = Không, 3 = Không chắc).
Dewormed	Con vật đã được tẩy giun (1 = Có, 2 = Không, 3 = Không chắc).
Sterilized	Con vật đã được diệt khuẩn/triệt sản (1 = Có, 2 = Không, 3 = Không chắc).
Health	Tình trạng sức khỏe (1 = Khỏe mạnh, 2 = Thương tật nhẹ, 3 = Thương tích nghiêm trọng, 0 = Không rõ).
Quantity	Số lượng con vật có trong hồ sơ.
Fee	Phí nhận nuôi (0 = Miễn phí).
State	Vị trí bang ở Malaysia.
RescuerID	ID của người cứu hộ.
VideoAmt	Tổng số video đã tải lên của con vật.
PhotoAmt	Tổng số ảnh đã tải lên của con vật.
Description	Hồ sơ mô tả của con vật. Ngôn ngữ chính được sử dụng là tiếng Anh, một số bằng tiếng Malaysia hoặc Trung Quốc.

## **Tốc độ nhận nuôi**

Tốc độ nhận nuôi là cột thể hiện con vật có được nhận nuôi hay không và tốc độ nhận nuôi như thế nào. Bất cứ khi nào con vật được liệt kê trong trang web 'PetFinder.my', PetFinder sẽ theo dõi thời gian đã trôi qua cho đến khi ai đó nhận nuôi. Các giá trị của cột tốc độ nhận nuôi được phân loại thành 5 mức:

- 0 - Con vật được nhận vào cùng ngày khi thông tin về nó được niêm yết.
- 1 - Con vật được nhận nuôi từ 1 đến 7 ngày (tuần đầu tiên) sau khi thông tin về nó được niêm yết.
- 2 - Con vật được nhận nuôi từ 8 đến 30 ngày (tháng đầu tiên) sau khi thông tin về nó được niêm yết.
- 3 - Con vật được nhận nuôi từ 31 đến 90 ngày (tháng thứ 2 & 3) sau khi thông tin về nó được niêm yết.
- 4 - Con vật không được nhận nuôi sau 100 ngày kể từ khi thông tin về nó được niêm yết.

Tốc độ nhận nuôi là biến cần dự đoán.

### **1.2.2. Hình ảnh**

Đối với con vật có ảnh, chúng sẽ được đặt tên theo định dạng PetID-ImageNumber.jpg. Với trường hợp ảnh hồ sơ của con vật là ảnh mặc định, vì mục đích riêng tư, khuôn mặt, số điện thoại và email đã được giấu đi.

### **1.2.3. Metadata ảnh**

Hình ảnh được chạy thông qua API Vision của Google, cung cấp phân tích về Chú thích khuôn mặt, chú thích nhãn, chú thích văn bản và thuộc tính của hình ảnh. Ta có thể tùy ý sử dụng thông tin bổ sung này để phân tích hình ảnh của mình.

Định dạng tên tệp là PetID-ImageNumber.json.

Một số thuộc tính sẽ không tồn tại trong tệp JSON nếu không có (ví dụ Chú thích khuôn mặt). Chú thích Văn bản đã được đơn giản hóa thành chỉ 1 mục nhập của toàn bộ mô tả văn bản (thay vì kết quả JSON chi tiết được chia nhỏ theo từng ký tự và từ). Số điện thoại và email đã được giấu đi trong chú thích văn bản.

Tham chiếu API Google Vision: <https://cloud.google.com/vision/docs/reference/rest/v1/images/annotate>

### **1.2.4. Dữ liệu mô tả**

Mô tả trong hồ sơ của từng con vật được đưa qua API ngôn ngữ tự nhiên của Google (Google's Natural Language API), cung cấp phân tích về tình cảm và các thực thể chính. Thông tin bổ sung này có thể được sử dụng để phân tích mô tả con vật.

Định dạng tên tệp là PetID.json

Tham chiếu API ngôn ngữ tự nhiên của Google: <https://cloud.google.com/natural-language/docs/basics>

## 2. THUỐC ĐO

### 2.1. Các thước đo phổ biến cho bài toán phân loại

Khi xây dựng một mô hình Machine Learning, chúng ta cần một phép đánh giá để xem mô hình sử dụng có hiệu quả không và để so sánh khả năng của các mô hình, thước đo sẽ giống như kim chỉ nam định hướng cách ta sẽ giải quyết xuyên suốt toàn bộ quá trình.

Như đã mô tả ở phần trước, đây là bài toán phân loại. Có rất nhiều cách đánh giá một mô hình phân lớp. Tuỳ vào những bài toán khác nhau mà chúng ta sử dụng các phương pháp khác nhau. Các phương pháp thường được sử dụng là: accuracy score, confusion matrix, ROC curve, Area Under the Curve, Precision and Recall, F1 score, Top R error.

#### 2.1.1. Accuracy

Cách đơn giản và hay được sử dụng nhất là accuracy (độ chính xác). Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử. Tuy nhiên accuracy chỉ phù hợp với các bài toán mà kích thước các lớp dữ liệu là tương đối như nhau.

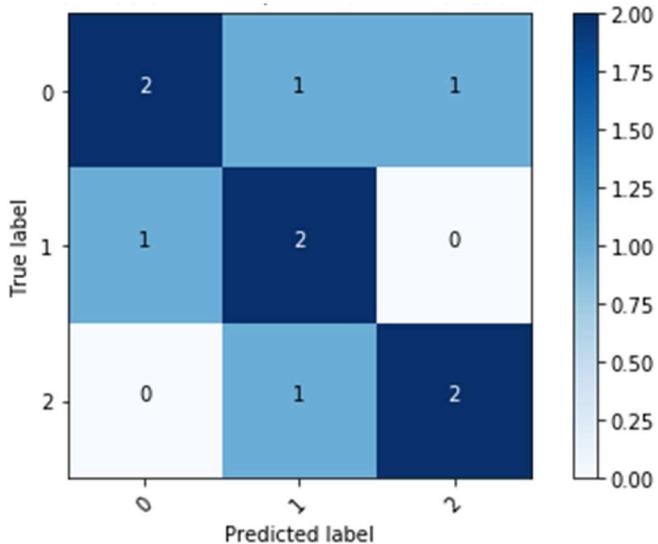
$$\text{Accuracy} = \frac{\text{Correct}}{\text{Total}}$$

Hình 2.1. Thước đo độ chính xác

#### 2.1.2. Confusion matrix

Cách tính sử dụng accuracy như ở trên chỉ cho chúng ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác. Để có thể đánh giá được các giá trị này, chúng ta sử dụng một ma trận được gọi là confusion matrix.

Về cơ bản, confusion matrix thể hiện có bao nhiêu điểm dữ liệu thực sự thuộc vào một class, và được dự đoán là rơi vào một class. Confusion matrix giúp có cái nhìn rõ hơn về việc các điểm dữ liệu được phân loại đúng/sai như thế nào.

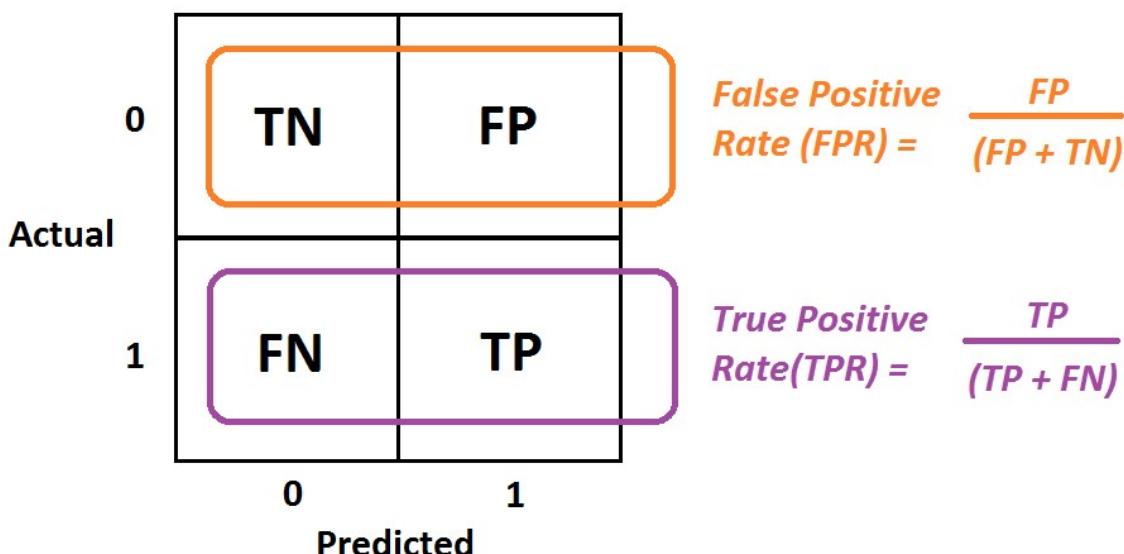


Hình 2.2. Confusion matrix trong bài toán phân loại

### 2.1.3. TPR, FPR, TNR, FNR

Cách đánh giá này thường được áp dụng cho các bài toán phân lớp có hai lớp dữ liệu. Cụ thể hơn, trong hai lớp dữ liệu này có một lớp quan trọng hơn lớp kia và cần được dự đoán chính xác. Ví dụ, trong bài toán xác định có bệnh ung thư hay không thì việc không bị sót (miss) quan trọng hơn là việc chẩn đoán nhầm âm tính thành dương tính. Trong bài toán xác định có mìn dưới lòng đất hay không thì việc bỏ sót nghiêm trọng hơn việc báo động nhầm rất nhiều. Hay trong bài toán lọc email rác thì việc cho nhầm email quan trọng vào thùng rác nghiêm trọng hơn việc xác định một email rác là email thường.

Trong những bài toán này, người ta thường định nghĩa lớp dữ liệu quan trọng hơn cần được xác định đúng là lớp Positive (P-dương tính), lớp còn lại được gọi là Negative (N-âm tính).



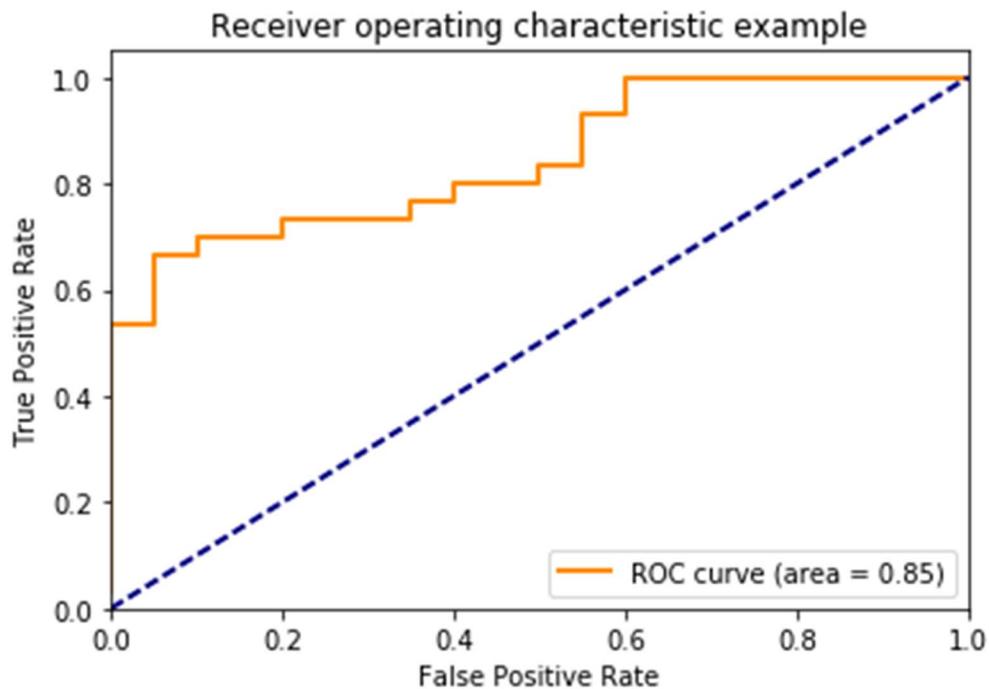
Hình 2.3. Chỉ số TPR, FPR trong phân loại 2 lớp

#### 2.1.4. ROC, AUC

Ứng với mỗi giá trị của ngưỡng, ta sẽ thu được một cặp (FPR, TPR). Biểu diễn các điểm (FPR, TPR) trên đồ thị khi thay đổi threshold từ 0 tới 1 ta sẽ thu được một đường được gọi là Receiver Operating Characteristic curve hay ROC curve.

Dựa trên ROC curve, ta có thể chỉ ra rằng một mô hình có hiệu quả hay không. Một mô hình hiệu quả khi có FPR thấp và TPR cao, tức tồn tại một điểm trên ROC curve gần với điểm có tọa độ (0, 1) trên đồ thị (góc trên bên trái). Curve càng gần thì mô hình càng hiệu quả.

Có một thông số nữa dùng để đánh giá được gọi là Area Under the Curve hay AUC. Đại lượng này chính là diện tích nằm dưới ROC curve màu cam. Giá trị này là một số dương nhỏ hơn hoặc bằng 1. Giá trị này càng lớn thì mô hình càng tốt.

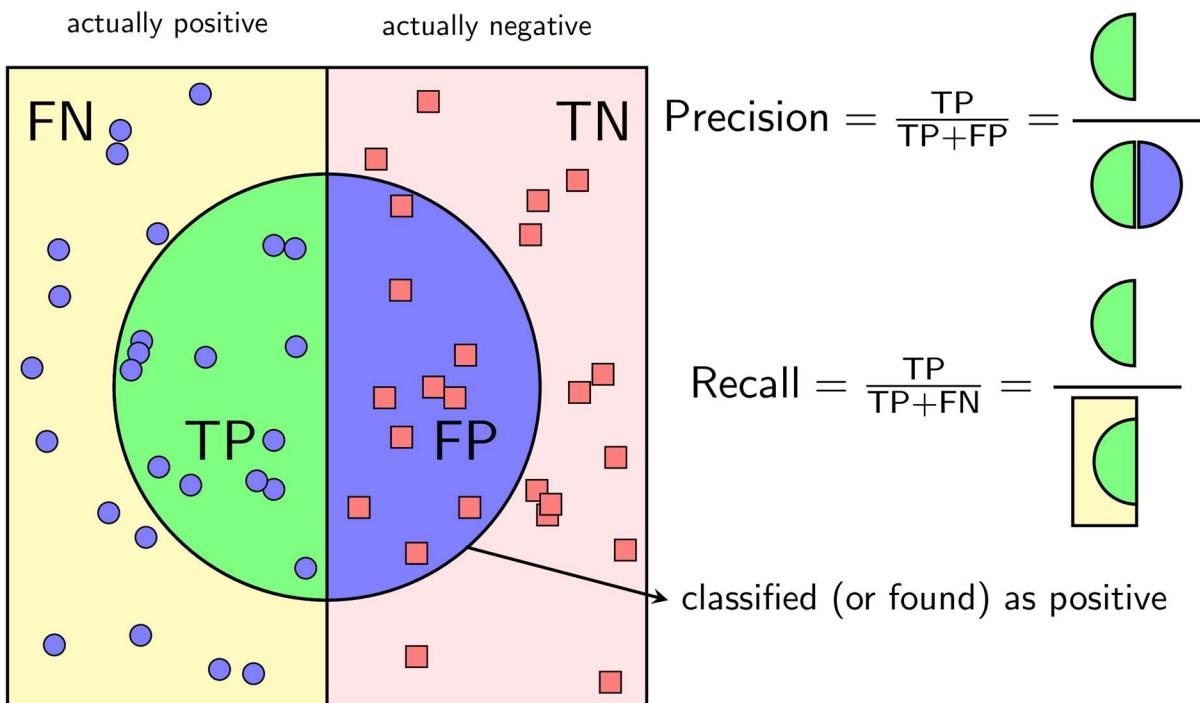


Hình 2.4. Đồ thị chỉ số ROC, AUC

#### 2.1.5. Precision và Recall

Với bài toán phân loại mà tập dữ liệu của các lớp là chênh lệch nhau rất nhiều, có một phép đo hiệu quả thường được sử dụng là Precision-Recall.

Trước hết xét bài toán phân loại nhị phân. Ta cũng coi một trong hai lớp là positive, lớp còn lại là negative. Precision cao đồng nghĩa với việc độ chính xác của các điểm tìm được là cao. Recall cao đồng nghĩa với việc True Positive Rate cao, tức tỉ lệ bỏ sót các điểm thực sự positive là thấp.



Hình 2.5. Cách tính precision và recall

## 2.2. Quadratic weighted kappa

Các thước đo được đề cập ở trên được dùng phổ biến để đánh giá mô hình phân lớp. Đặc điểm chung của các mô hình này là các lớp định danh không có quan hệ so sánh với nhau. Tuy nhiên với bài toán Pet finder này, mặc dù đầu ra là mô hình phân loại, tuy nhiên mỗi lớp lại có thứ tự so sánh được với nhau từ lớp 0 (là thời gian nhanh nhất) cho tới lớp 4 (là thời gian chậm nhất) nên nhóm đã quyết định sử dụng một thước đo ít phổ biến hơn cho mô hình phân loại nhưng phù hợp với trường hợp này là quadratic weighted kappa.

Kết quả của chỉ số kappa này thường thay đổi từ 0 (tương đương với việc lựa chọn hoàn toàn ngẫu nhiên) đến 1 (phân lớp đầu ra hoàn toàn chính xác). Kappa có trọng số bậc hai được tính giữa điểm số được mong đợi đã biết và điểm số được dự đoán.

Kết quả có 5 xếp hạng khả dĩ là 0, 1, 2, 3, 4. Kappa có trọng số bậc hai được tính như sau: Đầu tiên, một ma trận biểu đồ NxN O được xây dựng, sao cho  $O_{i,j}$  tương ứng với số lượng thực tế thuộc lớp i (thực tế) và nhận được dự đoán là lớp j. Một ma trận trọng số NxN, w, được tính toán dựa trên sự khác biệt giữa điểm phân loại thực tế và điểm phân loại dự đoán:

$$w_{i,j} = \frac{(i - j)^2}{(N - 1)^2}$$

```
w = np.zeros((5,5))
for i in range(5):
    for j in range(5):
        w[i][j] = (i-j)**2/16
w

array([[0.      , 0.0625, 0.25   , 0.5625, 1.      ],
       [0.0625, 0.      , 0.0625, 0.25   , 0.5625],
       [0.25   , 0.0625, 0.      , 0.0625, 0.25   ],
       [0.5625, 0.25   , 0.0625, 0.      , 0.0625],
       [1.      , 0.5625, 0.25   , 0.0625, 0.      ]])
```

Hình 2.6. Ma trận trọng số của chỉ số Kappa trong bài

Ma trận biểu đồ NxN của các lớp dự kiến, E, được tính toán, giả định rằng không có mối tương quan giữa các điểm xếp hạng. Đây được tính là tích ngoài giữa vectơ biểu đồ lớp thực tế và vectơ biểu đồ lớp được dự đoán, được chuẩn hóa sao cho E và O có tổng bằng nhau.

Tù ba ma trận này, kappa có trọng số bậc hai được tính như sau:

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}$$

### 3. PHƯƠNG PHÁP XỬ LÝ

#### 3.1. Sử dụng Functional API trong mạng học sâu

Thông thường khi tạo một mạng nơ-ron ta thường sử dụng Sequential API. Sequential API, như tên gọi của nó, cho phép ta tạo mô hình từng lớp theo kiểu từng bước.

Sequential API là cách dễ nhất để khởi tạo và chạy mô hình học sâu, nhưng tất nhiên cũng có giới hạn nhất định, Khi sử dụng Sequential API, ta không thể tạo các mô hình mà có khả năng:

- Chia sẻ layers
- Có nhánh (branches) (không thể hoặc rất khó khăn)
- Có nhiều đầu vào (multiple inputs)
- Có nhiều đầu ra (multiple outputs)

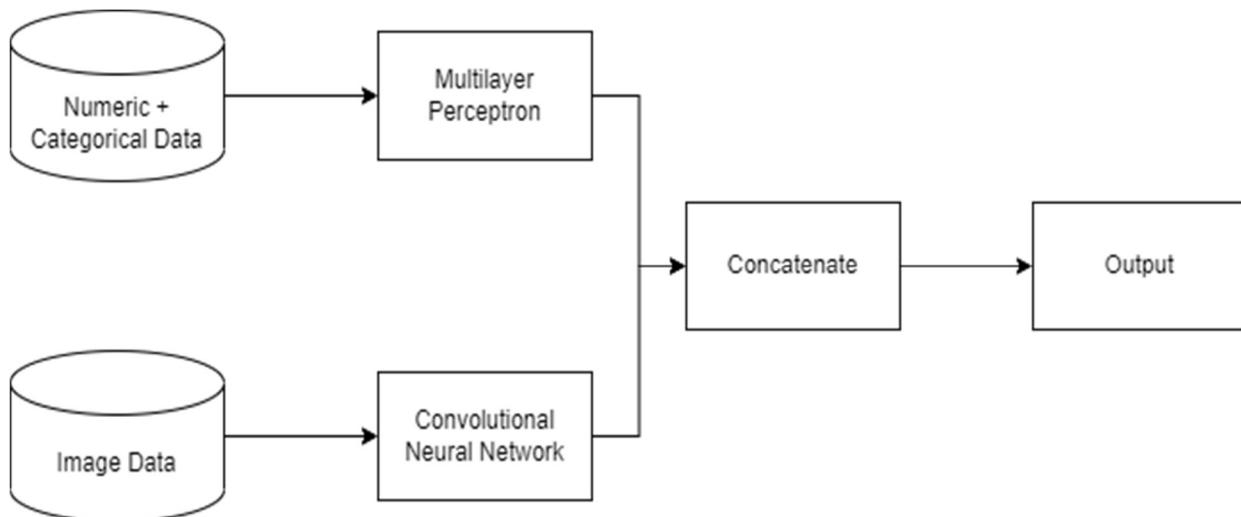


Hình 3.1. Ví dụ mô hình sử dụng Sequential API

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

Trong khi đó nhu cầu của mô hình trong bài là có nhiều đầu vào với đa dạng kiểu dữ liệu khác nhau nên cách sử dụng Sequential API thông thường không phù hợp. Ở đây ta có thể tìm đến phương án sử dụng Functional API, mà được khá nhiều người sử dụng. Functional API rất dễ sử dụng. Việc sử dụng Functional API cho phép:

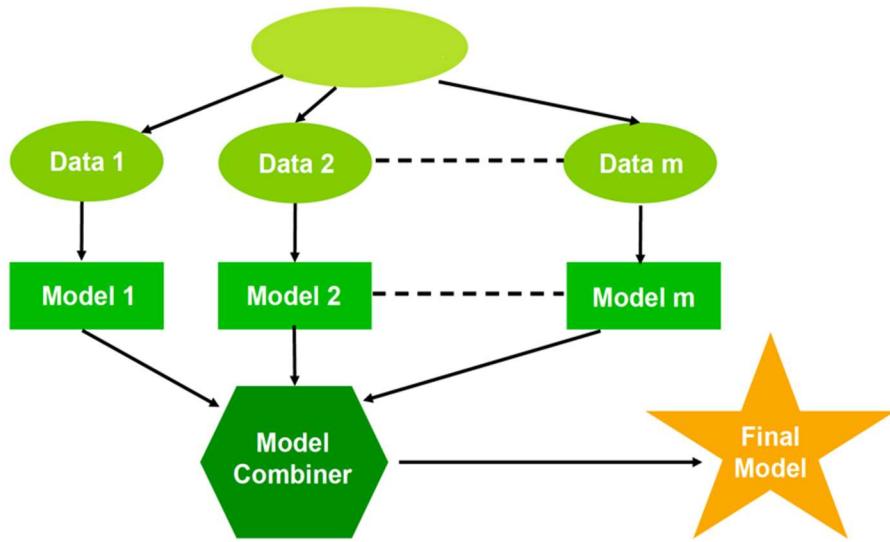
- Tạo mô hình phức tạp
- Đáp ứng được các bài toán nhiều đầu vào, nhiều đầu ra
- Dễ dàng định nghĩa các nhánh trong kiến trúc mô hình
- Thiết kế đồ thị xoay chiều có hướng (DAGs)
- Dễ dàng chia sẻ các lớp bên trong kiến trúc mô hình
- Đặc biệt là bất kì Sequential model nào cũng có thể triển khai bằng cách sử dụng Keras Functional API



Hình 3.2. Mô hình 2 kiểu dữ liệu đầu vào sử dụng Functional API

### **3.2. Trích chọn đặc trưng**

Do đặc thù của bài toán này là kiểu dữ liệu đầu vào rất đa dạng nên không thể chỉ sử dụng một mô hình cho tất cả các loại dữ liệu ngay từ đầu mà có thể sử dụng một hướng tiếp cận thứ hai là trích chọn các đặc trưng của từng loại dữ liệu đưa về dạng bảng và kết hợp các bảng lại thành một bảng tổng và áp dụng một mô hình cuối cùng trên bảng tổng đó.



Hình 3.3. Xử lý mỗi nguồn dữ liệu bằng một mô hình riêng trước khi kết hợp

#### 4. DỮ LIỆU METADATA ẢNH

Dữ liệu metadata ảnh được tạo ra bằng Cloud Vision API của Google Cloud. Tất cả các bức ảnh được đưa qua API và trích xuất ra các đặc trưng dưới dạng tệp Json. Các đặc trưng này có thể tham khảo tại tài liệu của Google Cloud theo đường dẫn <https://cloud.google.com/vision/docs/features-list>. Cụ thể trong bộ dữ liệu của bài có 5 đặc trưng của ảnh được trích xuất gồm có:

##### 4.1. faceAnnotations

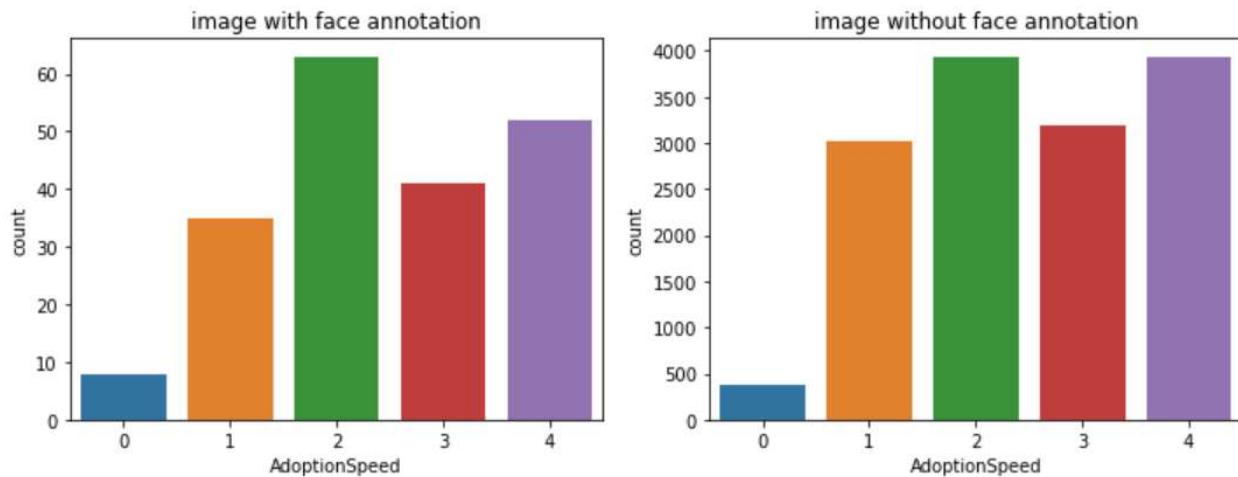
Tính năng Nhận diện khuôn mặt phát hiện nhiều khuôn mặt trong một hình ảnh cùng với các thuộc tính khuôn mặt chính liên quan, chẳng hạn như trạng thái cảm xúc hoặc đội mũ bảo hiểm. Nhận dạng khuôn mặt cá nhân cụ thể không được hỗ trợ tại API này.



Hình 4.1. Đặc trưng faceAnnotations

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cùu hộ với dữ liệu đa dạng

Bên dưới là biểu đồ phân bố thời gian được nhận nuôi của con vật theo ảnh có gương mặt chủ và ảnh không có gương mặt chủ.



Hình 4.2. Thời gian chờ nhận nuôi metadata faceAnnotation

### 4.2. LabelAnnotations

Vision API có thể phát hiện và trích xuất thông tin về các thực thể trong một hình ảnh, trên nhiều nhóm danh mục. Nhãn có thể xác định các đối tượng, vị trí, hoạt động chung, loài động vật, sản phẩm, v.v. Nếu cần các nhãn tùy chỉnh cho mục tiêu cụ thể, Cloud AutoML Vision cho phép đào tạo mô hình học máy tùy chỉnh để phân loại hình ảnh.

Nhãn chỉ được trả lại bằng tiếng Anh. Cloud Translation API có thể dịch các nhãn tiếng Anh sang bất kỳ ngôn ngữ nào.

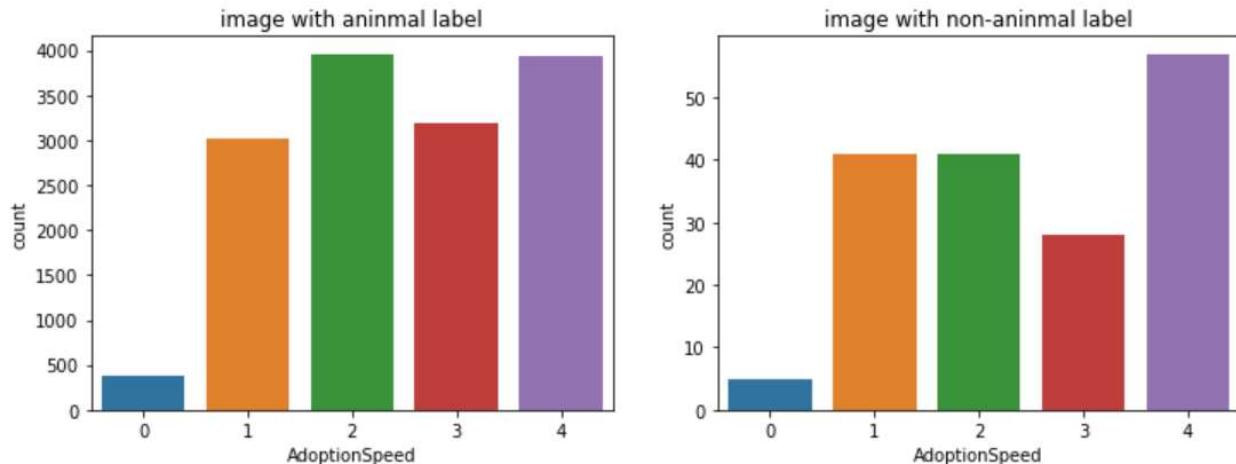


Description	Score
Street	0.872
Snapshot	0.852
Town	0.848
Night	0.804
Alley	0.713

Hình 4.3. Đặc trưng labelAnnotations

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

Thống kê từ tệp JSON ta có thể thấy tốc độ được nhận nuôi của những con vật có metadata labelAnnotations liên quan đến động vật nhanh hơn hẳn những con vật có metadata labelAnnotations không liên quan tới động vật

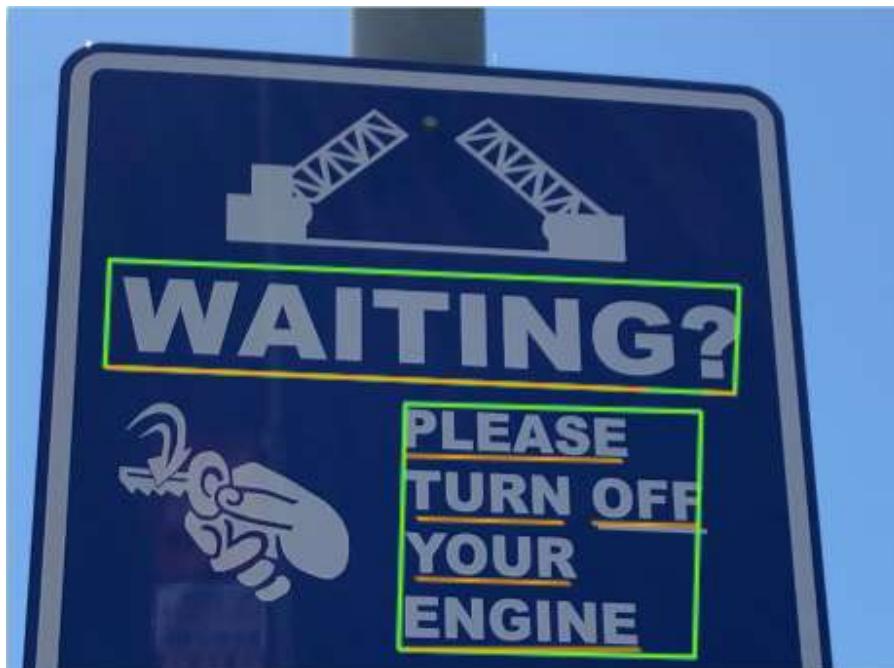


Hình 4.4. Thời gian chờ nhận nuôi theo nhãn metadata

### 4.3. TextAnnotations

API Vision có thể phát hiện và trích xuất văn bản từ hình ảnh. Có hai tính năng chú thích hỗ trợ nhận dạng ký tự quang học (OCR):

TEXT\_DETECTION phát hiện và trích xuất văn bản từ bất kỳ hình ảnh nào. Ví dụ: một bức ảnh có thể chứa biển báo đường phố hoặc biển báo giao thông. Tệp JSON bao gồm toàn bộ chuỗi được trích xuất, cũng như các từ riêng lẻ và các bounding box của chúng.



Hình 4.5. TEXT\_DETECTION OCR

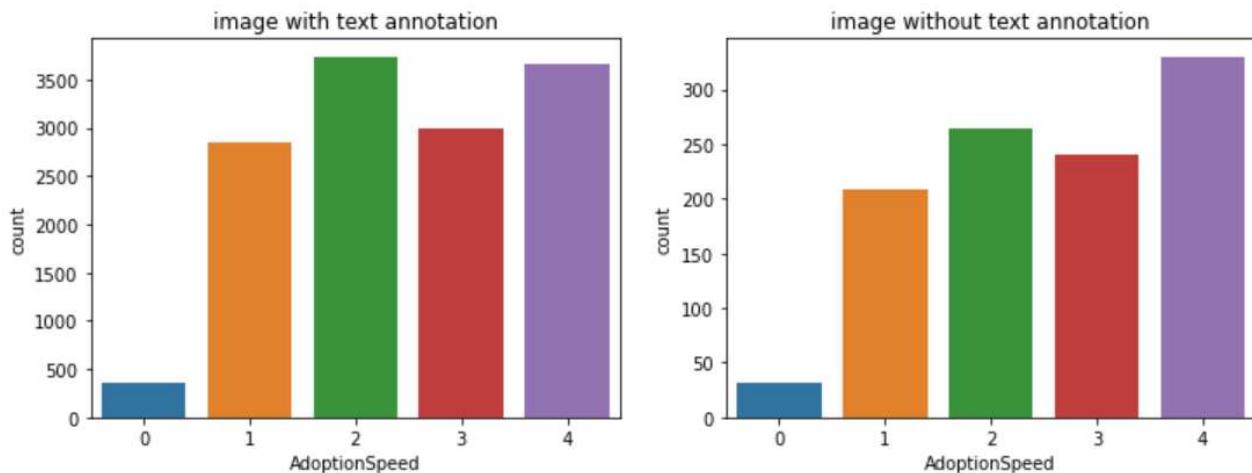
## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

DOCUMENT\_TEXT\_DETECTION cũng trích xuất văn bản từ hình ảnh, nhưng phản hồi được tối ưu hóa cho văn bản và tài liệu dày đặc. JSON bao gồm thông tin trang, khôi, đoạn, từ và ngắt.



Hình 4.6. DOCUMENT\_TEXT\_DETECTION OCR

Dưới đây là thống kê thời gian chờ nhận nuôi của các con vật có chú thích và không có chú thích trong ảnh. Những con vật trong ảnh có chú thích thường có thời gian chờ nhận nuôi ngắn hơn những con vật không có chú thích.



Hình 4.7. Thời gian chờ nhận nuôi theo metadata textAnnotation

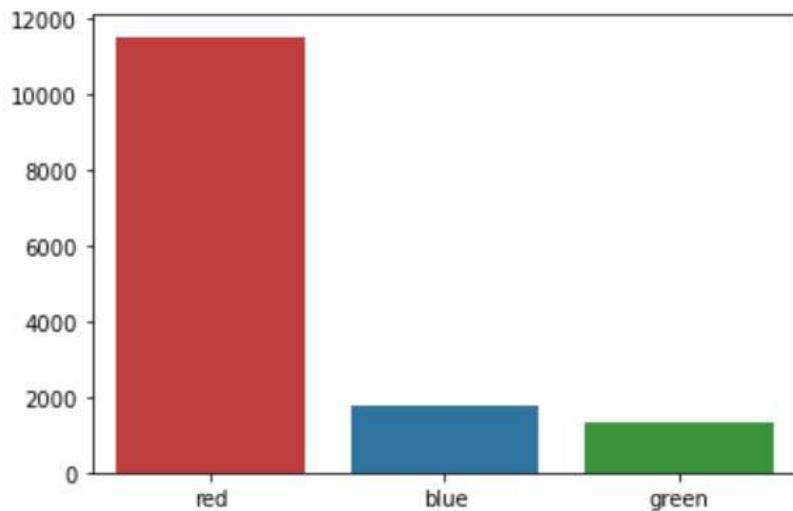
#### 4.4. ImagePropertiesAnnotation

Tính năng Thuộc tính Hình ảnh phát hiện các thuộc tính chung của hình ảnh, chẳng hạn như màu chủ đạo.



Hình 4.8. Đặc trưng imagePropertiesAnnotation

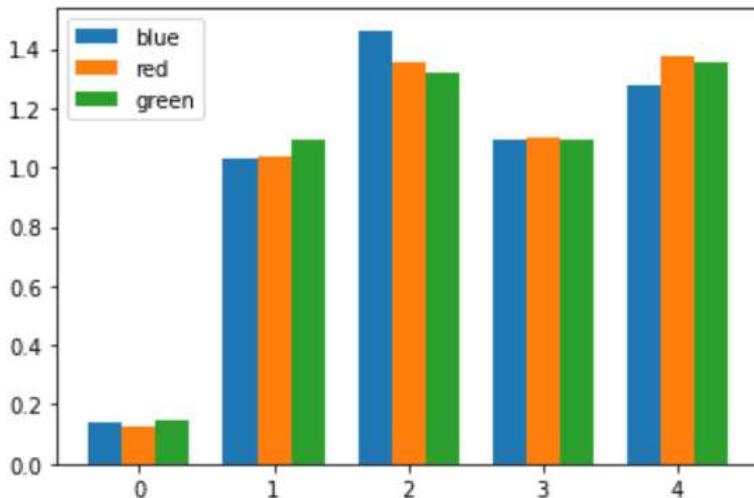
Đa số con vật có ảnh màu chủ đạo là đỏ với số lượng 11542 chiếm 78% số con vật, kế đến là màu xanh dương chiếm 12% và 10% màu xanh lá



Hình 4.9. Số lượng ảnh theo màu chủ đạo

Dưới đây là phân bố thời gian chờ nhận nuôi theo màu chủ đạo của ảnh. Về cơ bản số lượng ảnh có màu chủ đạo trong mỗi lớp không quá khác biệt, tuy nhiên những ảnh có màu chủ đạo là xanh lá hoặc xanh dương thường có thời gian chờ ngắn hơn những ảnh có màu chủ đạo là đỏ.

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng



Hình 4.10. Thời gian chờ nhận nuôi theo màu chủ đạo của ảnh

### 4.5. cropHintsAnnotation

Tính năng Crop Hint để xác định cho vùng cắt trên hình ảnh để tập trung vào vùng chứa thông tin chính.



Hình 4.11. Đặc trưng cropHintsAnnotation

Hầu như tất cả ảnh trong bài đều có chỉ số CropHint giữ nguyên kích thước, chỉ có gần 2.6% số ảnh được cắt giảm kích thước trên 10 pixel, sau khi phân tích, về cơ bản đặc trưng này chưa có đóng góp gì nhiều trong mô hình dự đoán.

## 5. DỮ LIỆU SENTIMENT

### 5.1. Phân tích cảm xúc

Phân tích cảm xúc cố gắng xác định thái độ tổng thể (tích cực hoặc tiêu cực) được thể hiện trong văn bản. Tình cảm được biểu thị bằng số điểm và giá trị độ lớn.

Các giá trị trường này được mô tả bên dưới:

- DocumentSentiment chứa tình cảm tổng thể của tài liệu, bao gồm các trường sau:
  - Score: Điểm của cảm xúc nằm trong khoảng từ -1,0 (tiêu cực) đến 1,0 (tích cực) và tương ứng với độ nghiêng cảm xúc tổng thể của văn bản.
  - Magnitud: Độ lớn cho biết sức mạnh tổng thể của cảm xúc (cả tích cực và tiêu cực) trong văn bản nhất định, trong khoảng từ 0,0 đến +inf. Không giống như điểm số, độ lớn không được chuẩn hóa, mỗi biểu hiện của cảm xúc trong văn bản (cả tích cực và tiêu cực) đều góp phần vào độ lớn của văn bản (vì vậy các khối văn bản dài hơn có thể có độ lớn lớn hơn).
- Language: Chứa ngôn ngữ của tài liệu, hoặc được chuyển trong yêu cầu ban đầu hoặc tự động được phát hiện nếu vắng mặt.
- Sentences: Chứa danh sách các câu được trích xuất từ tài liệu gốc, trong đó có:
  - Sentiment chứa các giá trị tình cảm cấp độ câu kèm theo mỗi câu, chứa các giá trị điểm và độ lớn như đã mô tả ở trên.

Điểm của cảm xúc của một tài liệu cho biết cảm xúc tổng thể của một tài liệu. Mức độ tình cảm của tài liệu cho biết mức độ nội dung tình cảm có trong tài liệu và giá trị này thường tỷ lệ với độ dài của tài liệu.

Điều quan trọng cần lưu ý là API ngôn ngữ tự nhiên (Nature Language API) chỉ ra sự khác biệt giữa cảm xúc tích cực và tiêu cực trong tài liệu, nhưng không xác định cảm xúc tích cực và tiêu cực cụ thể. *Ví dụ, "tức giận" và "buồn" đều được coi là cảm xúc tiêu cực. Tuy nhiên, khi API ngôn ngữ tự nhiên phân tích văn bản được coi là "tức giận" hoặc văn bản được coi là "buồn", phản hồi chỉ cho biết cảm xúc trong văn bản là tiêu cực, không phải "buồn" hoặc "tức giận".*

Một tài liệu có điểm trung lập (khoảng 0,0) có thể cho thấy một tài liệu ít cảm xúc hoặc có thể cho thấy nhiều cảm xúc lẫn lộn, với cả giá trị tích cực và tiêu cực cao đều loại bỏ mỗi tài liệu. Nói chung, bạn có thể sử dụng các giá trị độ lớn để phân biệt các trường hợp này, vì các tài liệu thực sự trung tính sẽ có giá trị độ lớn thấp, trong khi các tài liệu hỗn hợp sẽ có giá trị độ lớn cao hơn.

Khi so sánh các tài liệu với nhau (đặc biệt là các tài liệu có độ dài khác nhau), cần thiết sử dụng các giá trị độ lớn để hiệu chỉnh điểm đánh giá, vì chúng có thể giúp ta đánh giá lượng nội dung cảm xúc có liên quan.

Biểu đồ dưới đây cho thấy một số giá trị mẫu và cách giải thích chúng:

Sentiment	Sample Values
Clearly Positive*	"score": 0.8, "magnitude": 3.0
Clearly Negative*	"score": -0.6, "magnitude": 4.0
Neutral	"score": 0.1, "magnitude": 0.0
Mixed	"score": 0.0, "magnitude": 4.0

*Hình 5.1. Diễn giải mức độ cảm xúc theo các giá trị mẫu*

Cảm nhận “Tích cực rõ ràng” (clearly positive) và “Tiêu cực rõ ràng (clearly negative)” khác nhau đối với các trường hợp sử dụng và đối tượng khác nhau. Do vậy, tùy tình huống áp dụng mà ta nên xác định ngưỡng phù hợp và sau đó điều chỉnh ngưỡng sau khi thử nghiệm và xác minh kết quả. *Ví dụ: Ta có thể xác định ngưỡng của bất kỳ điểm nào trên 0,25 rõ ràng là tích cực, sau đó sửa đổi ngưỡng điểm thành 0,15 sau khi xem xét dữ liệu và kết quả và nhận thấy rằng điểm từ 0,15 - 0,25 cũng được coi là tích cực.*

## 5.2. Phân tích thực thể

Phân tích thực thể cung cấp thông tin về các thực thể trong văn bản, thường đề cập đến các “sự vật” được đặt tên như cá nhân nổi tiếng, địa danh, đồ vật thông thường,...

Thực thể thường chia thành hai loại: danh từ riêng chỉ các thực thể duy nhất (người, địa điểm cụ thể, ..) hoặc danh từ chung (còn được gọi là “danh nghĩa” trong xử lý ngôn ngữ tự nhiên). Một nguyên tắc chung cần tuân theo là nếu một cái gì đó là một danh từ, thì nó đủ điều kiện là một “thực thể”.

Lưu ý rằng API ngôn ngữ tự nhiên trả về các thực thể cho “Lawrence of Arabia” (phim) và “T.E. Lawrence” (người). Phân tích thực thể rất hữu ích để phân biệt các thực thể tương tự như “Lawrence” trong trường hợp này.

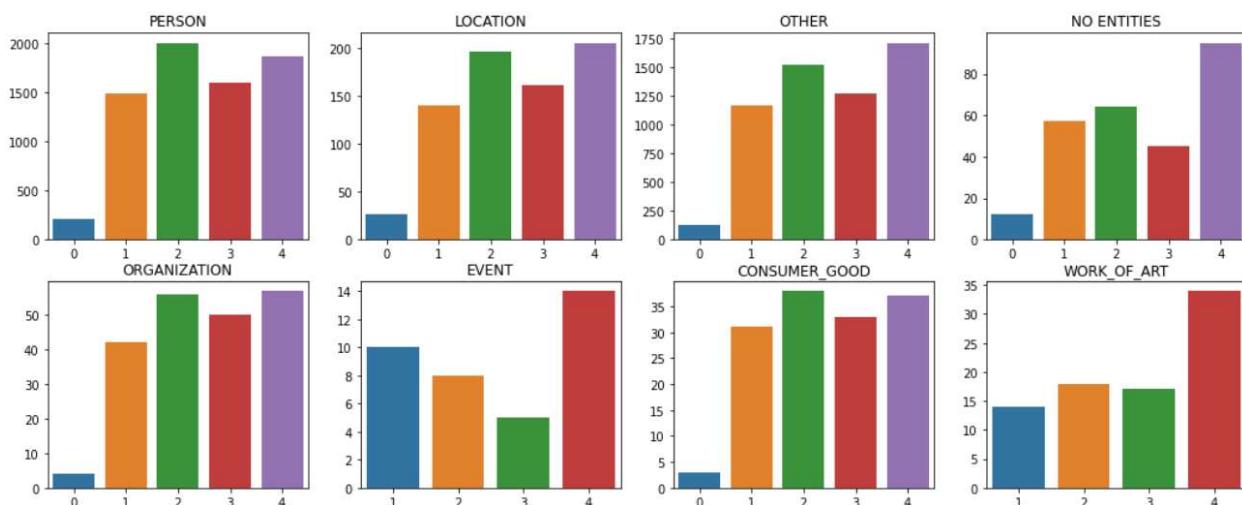
Các trường được sử dụng để lưu trữ các tham số của thực thể được liệt kê bên dưới:

- Type cho biết loại thực thể này (ví dụ: nếu đối tượng là người, địa điểm, hàng tiêu dùng, ...) Thông tin này giúp phân biệt và/hoặc phân biệt các thực thể và có thể được sử dụng để viết các mẫu hoặc trích xuất thông tin. *Ví dụ: giá trị kiểu có thể giúp phân biệt các thực thể có tên tương tự như “Lawrence of Arabia”, được gắn thẻ là WORK\_OF\_ART (phim), với “T.E. Lawrence”, được gắn thẻ là PERSON chẳng hạn.*
- Metadata chứa thông tin nguồn về kho tri thức của thực thể Kho lưu trữ bổ sung có thể được hiển thị trong tương lai. Trường này có thể chứa các trường con sau:
  - Wikipedia\_url (nếu có) chứa URL Wikipedia liên quan đến thực thể này.
  - Mid (nếu có) chứa số nhận dạng do máy tạo (MID) tương ứng với mục nhập Google Knowledge Graph của thực thể. Lưu ý rằng các giá trị giữa vẫn là

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

duy nhất trên các ngôn ngữ khác nhau, vì vậy bạn có thể sử dụng các giá trị đó để liên kết các thực thể với nhau từ các ngôn ngữ khác nhau. Để kiểm tra các giá trị MID này, vui lòng tham khảo tài liệu API Tìm kiếm Google Knowledge Graph.

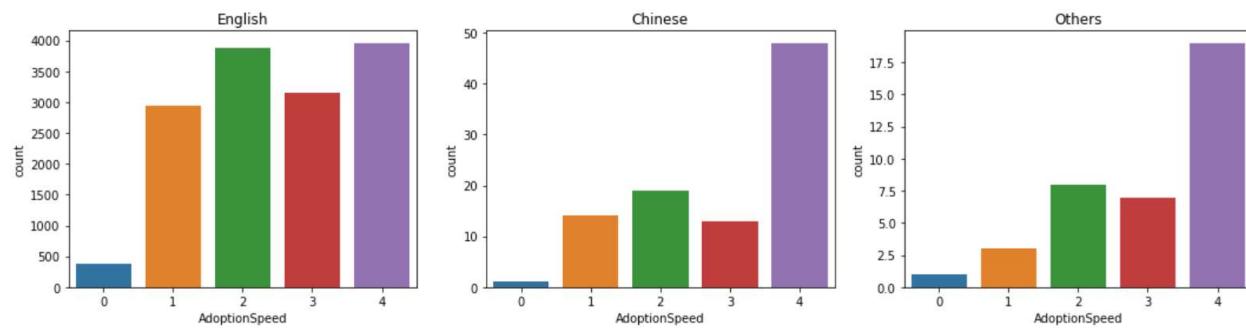
- Salience chỉ ra tầm quan trọng hoặc mức độ liên quan của thực thể này đối với toàn bộ nội dung tài liệu. Điểm số này có thể hỗ trợ việc truy xuất và tóm tắt thông tin bằng cách ưu tiên các thực thể nổi bật. Điểm gần 0,0 ít quan trọng hơn, trong khi điểm gần 1,0 lại rất quan trọng.
- Mentions chỉ ra các vị trí bù đắp trong văn bản nơi một thực thể được đề cập. Thông tin này có thể hữu ích nếu bạn muốn tìm tất cả các đề cập đến người “Lawrence” trong văn bản nhưng không phải là tiêu đề phim. Ta cũng có thể sử dụng các đề cập để thu thập danh sách các bí danh thực thể, chẳng hạn như “Lawrence”, đề cập đến cùng một thực thể “T.E. Lawrence”. Đề cập thực thể có thể là một trong hai loại: PROPER hoặc COMMON. *Ví dụ, một danh từ riêng Thực thể cho "Lawrence of Arabia", có thể được đề cập trực tiếp như tiêu đề phim hoặc như một danh từ chung ("tiểu sử phim" của T.E. Lawrence).*



Hình 5.2. Phân bổ các nhóm theo thực thể được xác định trong mô tả

Ngoài ra sentiment còn cung cấp thông tin về ngôn ngữ sử dụng trong phần mô tả. Từ biểu đồ dưới đây, ta có thể thấy rõ rằng những con vật có phần mô tả bằng tiếng Trung hay ngôn ngữ khác ngoài tiếng Anh có khả năng được nhận nuôi sớm thấp hơn hẳn, đây sẽ là một biến có giá trị dự báo tốt về khả năng được nhận nuôi của con vật.

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng



Hình 5.3. Thời gian chờ theo ngôn ngữ trong phần mô tả

## 6. PHÂN TÍCH DỮ LIỆU DẠNG BẢNG

### 6.1. Khai phá và xử lý dữ liệu missing

#### 6.1.1. Tổng quan về bộ dữ liệu

Type	Name	Age	Breed1	Breed2	Gender	...	RescuerID	VideoAmt	Description	PetID	PhotoAmt	AdoptionSpeed
0	2	Nibble	3	299	0	1 ...	8480853f516546f6cf33aa88cd76c379	0	Nibble is a 3+ month old ball of cuteness. He ...	86e1089a3	1.0	2
1	2	No Name Yet	1	265	0	1 ...	3082c7125d8fb66f7dd4bf4192c8b14	0	I just found it alone yesterday near my apartm...	6296e909a	2.0	0
2	1	Brisco	1	307	0	1 ...	fa90fa5b1ee11c86938398b60abc32cb	0	Their pregnant mother was dumped by her irresp...	3422e4906	7.0	3
3	1	Miko	4	307	0	2 ...	9238e4f44c71a75282e62f7136c6b240	0	Good guard dog, very alert, active, obedience ...	5842ff5	8.0	2
4	1	Hunter	1	307	0	1 ...	95481e953f8aed9ec3d16fc4509537e8	0	This handsome yet cute boy is up for adoption....	850a43f90	3.0	2
5	2	NaN	3	266	0	2 ...	22fe332bf9c924d4718005891c63fbed	0	This is a stray kitten that came to my house. ...	d24c30b4b	2.0	2
6	2	BULAT	12	264	264	1 ...	1e0b5a458b5b77f5af581d57ebf570b3	0	anyone within the area of ipoh or taiping who ...	1caa6fcdb	3.0	1

Type	Name	Age	Breed1	Breed2	Gender	...	RescuerID	VideoAmt	Description	PetID	PhotoAmt	AdoptionSpeed
14985	1	Terry	24	179	307	1 ...	719987dce7aeb027dfa91b480800199	0	been at my place for a while...am hoping to fin...	e7f7066b6	0.0	4
14986	2	Pets + Strays : BlueEyes BlackWhite	1	266	0	2 ...	90569c3f7cb0af35cba5dac82c0ac9d7	0	1 month old white + grey kitten for adoption n...	36e7f8d83	1.0	3
14987	1	Snowy	6	195	0	2 ...	79309f4027f2fedb4349a298c69fe56f	0	oooooo	4d163b731	1.0	0
14988	2	NaN	2	266	0	3 ...	61c84bd7bcb6fb31d2d480b1bcf9682e	0	I have 4 kittens that need to be adopt urgentl...	dc0935a84	3.0	2
14989	2	Serato & Eddie	60	265	264	3 ...	1d5096c4a5e159a3b750c5cf6ceabf	0	Serato(female cat- 3 color) is 4 years old and...	a01ab5b30	3.0	4
14990	2	Monkies	2	265	266	3 ...	6f40a7acfad5cc0bb3e44591ea446c05	0	Mix breed, good temperament kittens. Love huma...	d981b6395	5.0	3
14991	2	Ms Daym	9	266	0	2 ...	c311c0c569245baa147d91fa4e351ae4	0	she is very shy..adventures and independent...s...	e4da1c9e4	3.0	4
14992	1	Fili	1	307	307	1 ...	9ed1d5493d223eaa5024c1a031dbc9c2	0	Fili just loves laying around and also loves b...	a83d95ead	1.0	3

Hình 6.1. Kiểm tra các quan sát đầu và cuối trong bộ dữ liệu

### **1.2.2. Kiểm tra và xử lý các giá trị null (rỗng) trong bộ dữ liệu**

```
Type          0
Name         1257
Age           0
Breed1        0
Breed2        0
Gender         0
Color1        0
Color2        0
Color3        0
MaturitySize   0
FurLength      0
Vaccinated     0
Dewormed       0
Sterilized      0
Health          0
Quantity        0
Fee             0
State           0
RescuerID       0
VideoAmt        0
Description     12
PetID           0
PhotoAmt        0
AdoptionSpeed    0
dtype: int64
```

*Hình 6.2. Kiểm tra các giá trị null (rỗng) trong bộ dữ liệu*

#### **\* Nhận xét:**

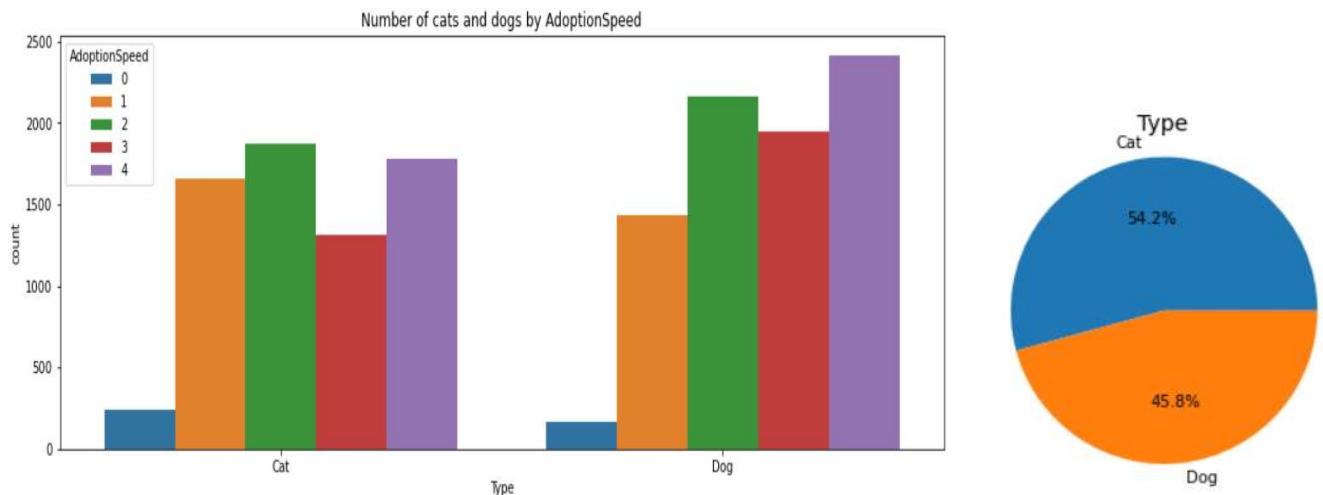
Các giá trị rỗng sẽ làm giảm độ chính xác khi đo đếm và vẽ sơ đồ cho dữ liệu. Qua kiểm tra ta thấy giá trị null nằm trong các trường Name, Description. Cụ thể:

- Trong cột Name chứa 1257/14993 bản ghi có dữ liệu rỗng, tương tự ở cột Description có chứa 12/14993 bản ghi có dữ liệu rỗng. Nhận thấy ở cả 2 trường này đều có kiểu dữ liệu là string, phụ thuộc vào điều tra thu thập mà có thể có giá trị hay không có gì trị, do vậy các giá trị rỗng vẫn có thể có ý nghĩa trong quyết định cuối cùng của bài toán.

**Hướng xử lý:** Với các ô dữ liệu chứa giá trị null trong cả 2 cột Name và Description ta thay thế bởi giá trị text trống (“ ”)

## 6.2. Phân tích biến đơn và xử lý dữ liệu

### 6.2.1. Biến Type



*Hình 6.3. Biểu đồ biểu diễn số lượng chó, mèo trong bộ dữ liệu*

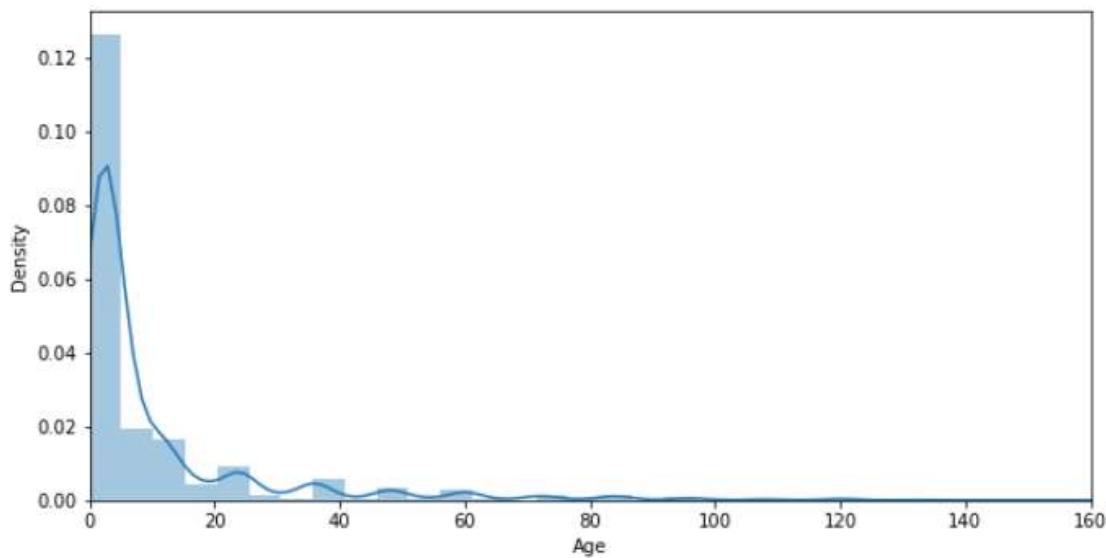
#### \* Nhận xét:

Số lượng chó chiếm tỉ lệ 45.8% trong bộ dữ liệu trong khi số lượng mèo chiếm tỉ lệ 54.2% trong bộ dữ liệu là khá cân bằng. Theo như biểu đồ phân loại theo thời gian được nhận nuôi của chó và mèo ta thấy:

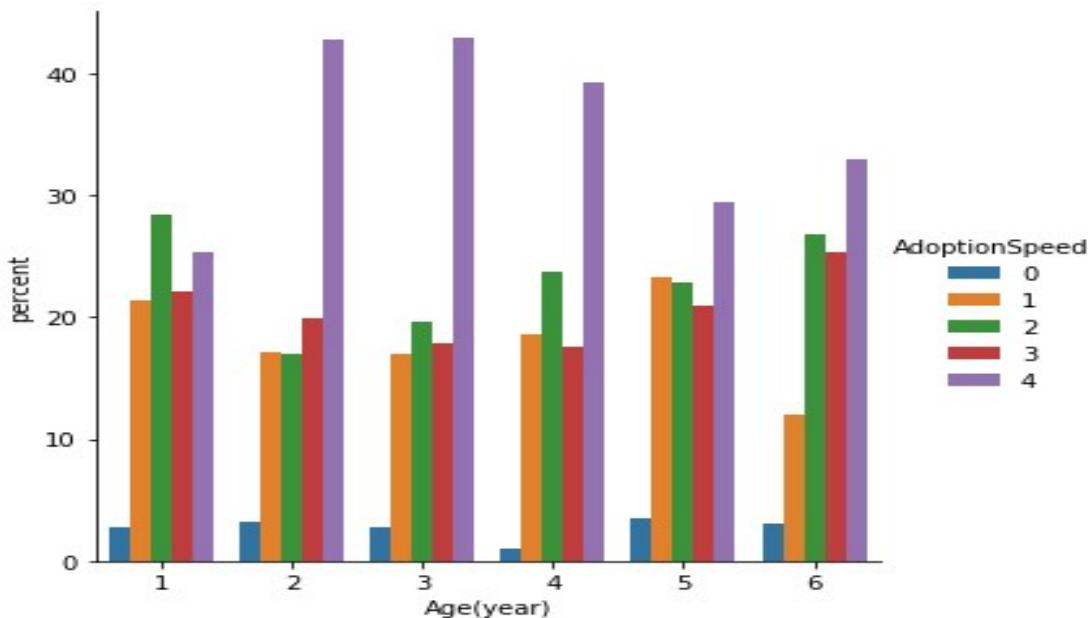
- Cả chó và mèo thời gian nhận nuôi của 2 con vật này phân bố thường trong khoảng thời gian từ 8 đến 30 ngày và sau 100 ngày kể từ ngày được nhận về trung tâm cứu hộ động vật.
- Tỉ lệ mèo được nhận nuôi trong tuần đầu tiên kể từ khi được nhận về lớn hơn so với tỉ lệ chó. Trong khi đó tỉ lệ chó được nhận nuôi khoảng thời gian sau 100 ngày là rất lớn.

Tóm lại, qua phân tích ta có thể thấy đa số mèo được nhận nuôi sớm hơn so với chó.

### 6.2.2. Biểu đồ Age



*Hình 6.4. Biểu đồ tuổi của vật nuôi theo tháng*

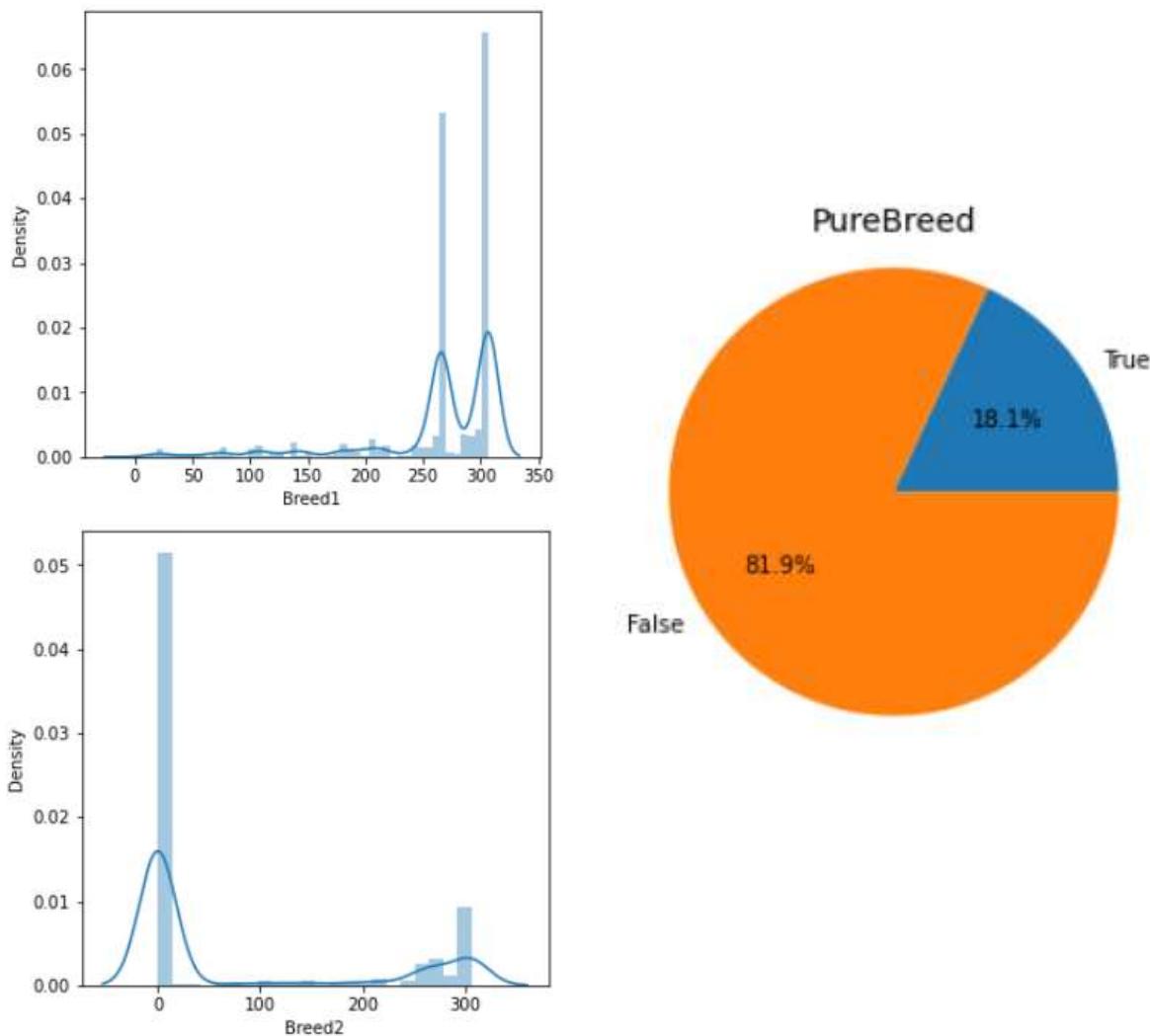


*Hình 6.5. Biểu đồ tuổi của vật nuôi theo năm*

#### \* Nhận xét:

- Đa số các con vật nuôi đều ở độ tuổi dưới 1 năm, số lượng vật nuôi có độ tuổi từ 3 năm trở lên là rất ít trong bộ dữ liệu.
- Các con vật nuôi có độ tuổi dưới 1 năm có thời gian nhận nuôi dưới 1 tháng trong khi các con vật có độ tuổi trên 1 năm tuổi có thời gian nhận nuôi thường trên 100 ngày.

### **6.2.3. Biến giống loài (Breed)**

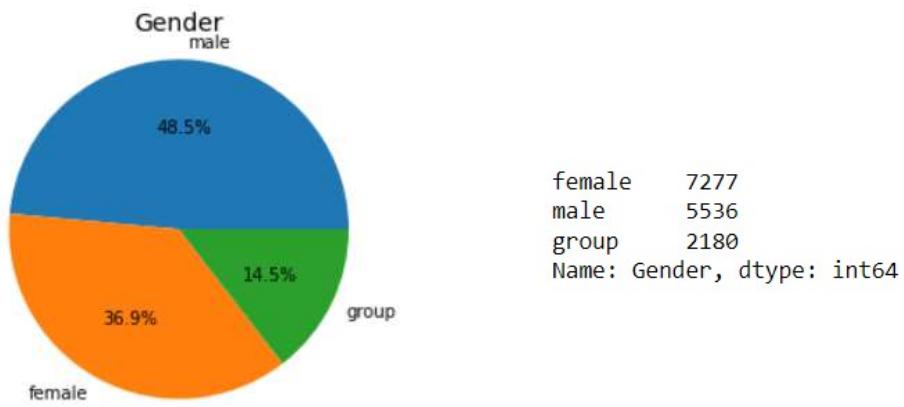


*Hình 6.6. Biểu đồ biểu diễn phân phối của 2 biến về giống loài*

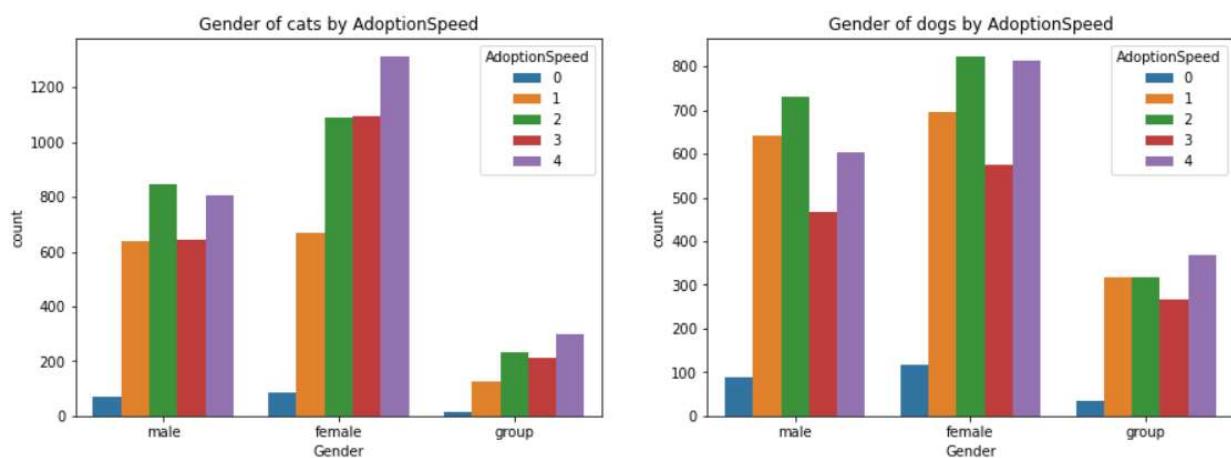
#### **Nhận xét:**

- Biến giống loài được biểu diễn bởi 2 trường dữ liệu gồm Breed1 (Giống loài của đời cha), Breed2 (giống loài của đời mẹ). Ta thấy cả 2 biến đều phân bố rời rạc trong khoảng từ 0 – 350 và tập trung nhiều ở các loài mang số hiệu khoảng 270 đến 300. Các giá trị 0 biểu thị vật nuôi có đời bố mẹ giống nhau.
- Việc có thông số về giống loài của đời trước giúp ta có thể xác định được vật nuôi có thuần chủng hay không. Qua phân tích ta thấy có 81.9% con vật được nhận về là không thuần chủng và 18.1% các con vật được nhận nuôi về thuần chủng

### **6.2.4. Biến Gender**



Hình 6.7. Biểu đồ biểu diễn tỉ lệ giới tính của vật nuôi

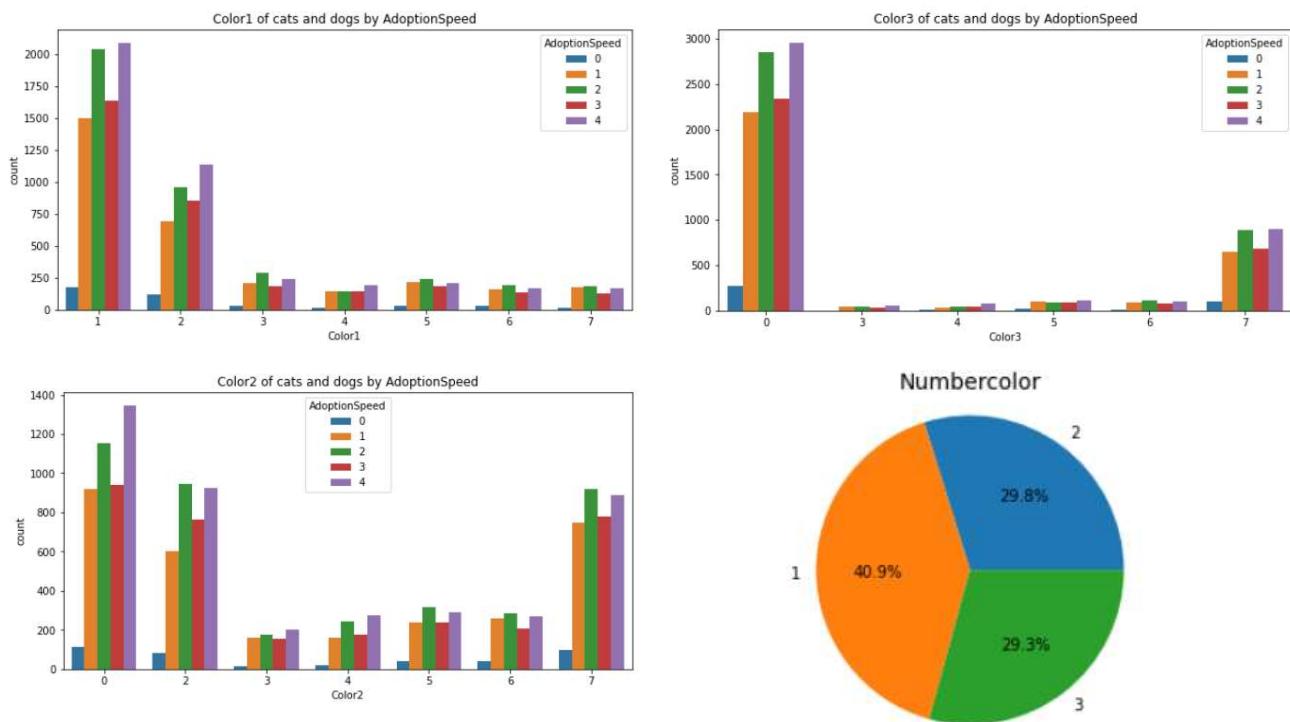


Hình 6.8. Biểu đồ biểu diễn tỉ lệ giới tính của vật nuôi theo thời gian được nhận nuôi

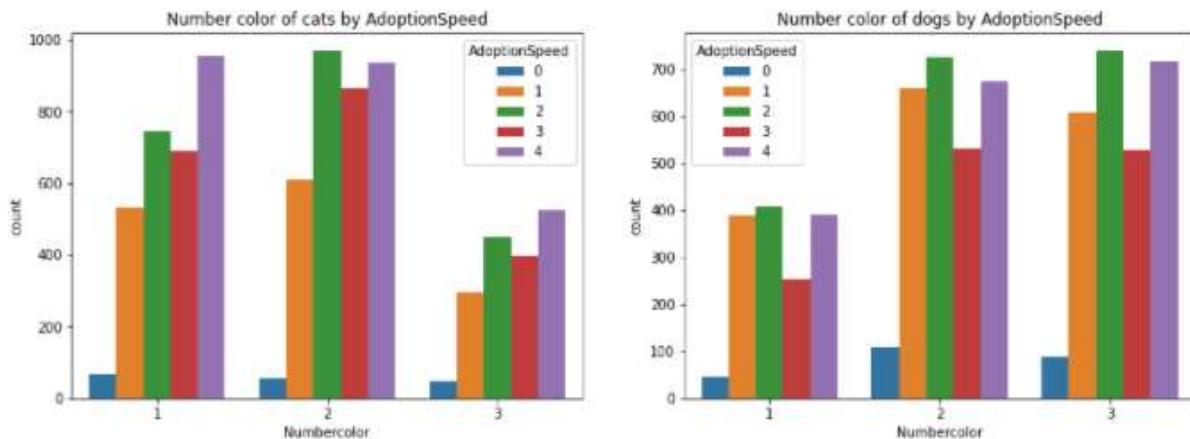
**Nhận xét:** Trong số các con vật được nhận nuôi số lượng con vật giống cái chiếm đa số với tỉ lệ 48.5%, tỉ lệ các con vật giống đực là 36.9%, trong khi đó có 14.5% là các bản ghi chưa 1 nhóm các con vật không xác định giới tính từng con.

- Xét riêng cho mèo ta thấy các con cái có thời gian nhận nuôi khá cân bằng nhau trong khi với các con đực và các nhóm các con mèo, thời gian nhận nuôi hầu hết là sau ít nhất 1 tháng.
- Đối với những con chó được nhận nuôi ta thấy thời gian nhận nuôi của các con đực và cái là như nhau. Với nhóm các con chó được nhận nuôi, tỉ lệ các con chó được nhận nuôi sau 100 ngày chiếm tỉ lệ cao nhất

### 6.2.5. Biểu màu lông (color1, color2, color3)



Hình 6.9. Biểu đồ biểu diễn các loại màu lông



Hình 6.10. Biểu đồ biểu diễn số màu lông theo thời gian được nhận nuôi

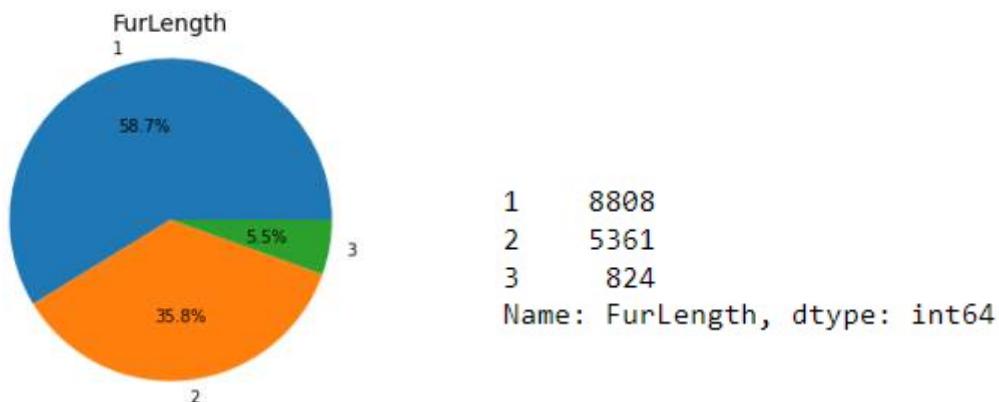
\* **Nhận xét:** Có 3 loại màu lông cơ bản được thống kê trong bộ dữ liệu với cường độ các màu khác nhau phân bố trong dải từ 0 đến 8. Về tổng thể ta thấy ở cả 3 màu cơ bản, vật nuôi đa phần ở các gam màu nhạt trong bảng thư viện màu sắc. Cùng với đó, các con vật nuôi có gam màu đậm hơn hoặc nhạt hơn thường có thời gian nhận nuôi từ 1-2 tháng hoặc trên 3 tháng. Các con vật nuôi có cường độ gam màu trung bình không có sự khác biệt rõ ràng với thời gian nhận nuôi.

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

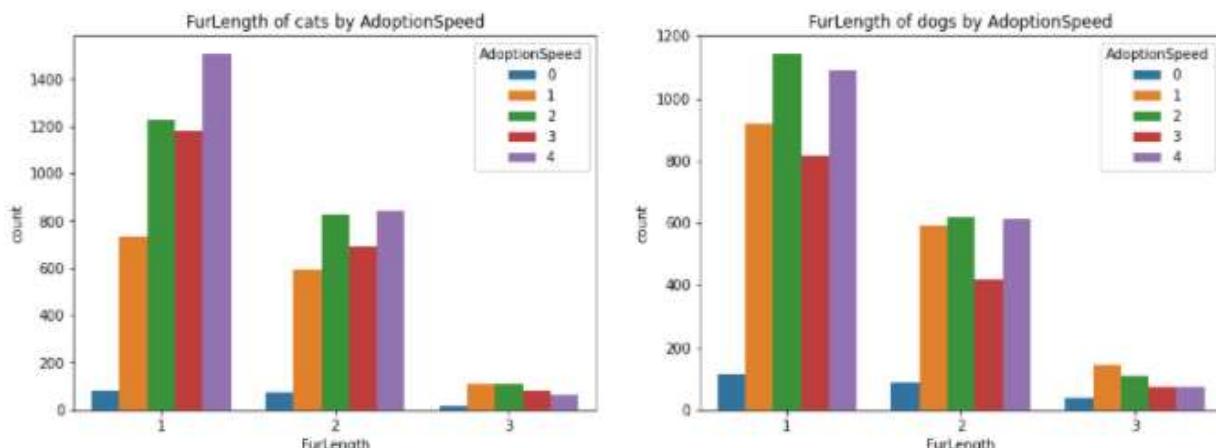
- Trong tập dữ liệu có 40.9% các con vật nuôi chỉ có 1 màu duy nhất, 29.8% các con vật nuôi có 2 màu trên cơ thể và có 29.3% các con vật nuôi có từ 3 màu trở lên.

- Với những con chó có 1 và 3 màu lông thường được nhận nuôi khoảng thời gian trên 100 ngày. Với những con mèo, thời gian nhận nuôi ở tất cả các loại màu lông đều rơi vào tuần ngay đầu tiên.

### **6.2.6. Biến độ dài lông (FurLength)**



Hình 6.11. Biểu đồ biểu diễn độ dài lông



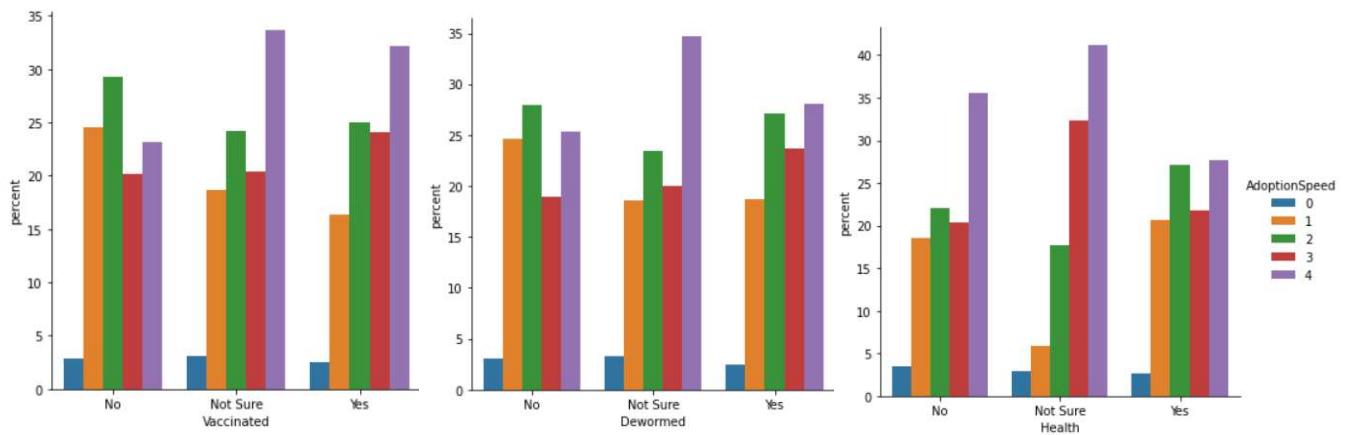
Hình 6.12. Biểu đồ biểu diễn độ dài lông theo thời gian được nhận nuôi

\* **Nhận xét:** Phần lớn các con vật được nhận nuôi có lông ngắn chiếm tỉ lệ 58.7%, số lượng con vật có lông dài chiếm rất ít chỉ 5.5%.

- Ở cả chó và mèo, các con vật có lông dài thường được nhận nuôi sớm hơn các con vật có lông ngắn hơn.
- Đặc biệt những con chó có lông ngắn thời gian được nhận nuôi đa số trên 100 ngày. Trong khi những con mèo có lông ngắn thường được nhận nuôi ở khoảng thời gian từ 1 tháng trở lại.

### **6.2.7 Các biến liên quan tới sức khỏe vật nuôi (Vaccinated, Dewormed, Health)**

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cưu hộ với dữ liệu đa dạng

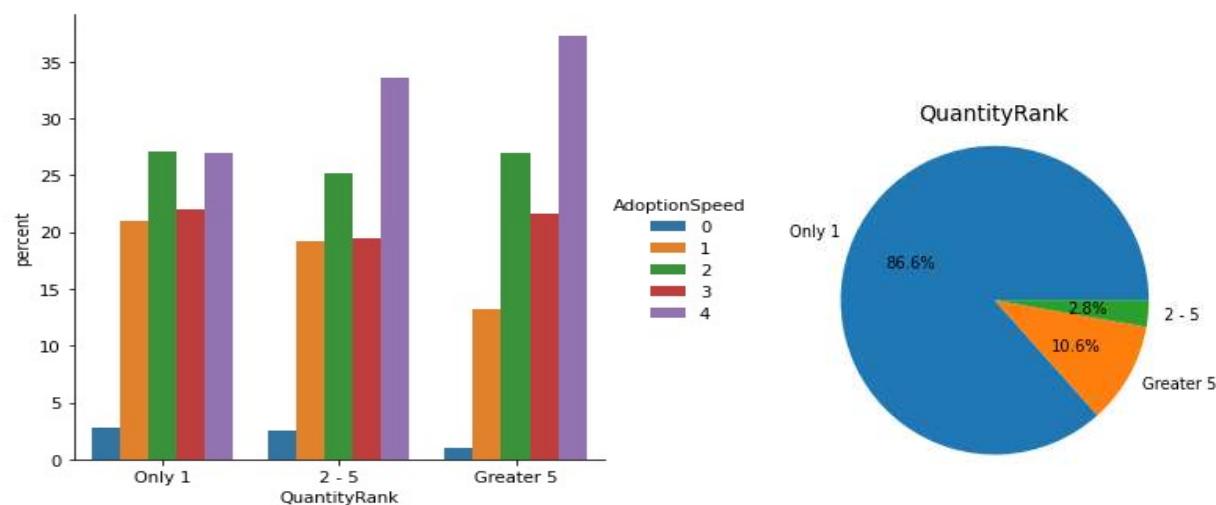


Hình 6.13. Biểu đồ tình trạng y tế các con vật nuôi theo thời gian được nhận nuôi

### Nhận xét:

- Hầu hết các con vật nuôi được nhận về trong tình trạng khỏe mạnh và đã được tẩy giun đầy đủ. Tỉ lệ các con vật đã được tiêm vaccine và chưa được tiêm tương đối cân bằng nhau.
- Với những con vật đã được tiêm vaccine và đã được tẩy giun thời gian được nhận nuôi là khoảng trên 100 ngày trong khi những con vật chưa được tiêm vaccine và chưa được tẩy giun thời gian nhận nuôi là khoảng dưới 1 tháng.
- Tỉ lệ các con vật có sức khỏe bình thường có thời gian được nhận nuôi ngắn hơn với các con vật có sức khỏe không tốt.

### 6.2.8. Số lượng con vật (Quantity)



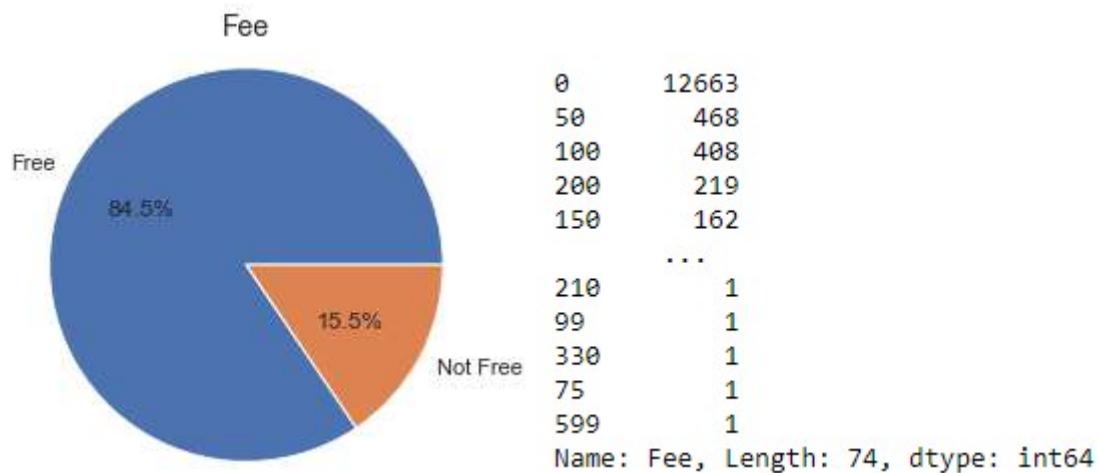
Hình 6.14. Biểu đồ biểu diễn số lượng vật nuôi hiện có

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

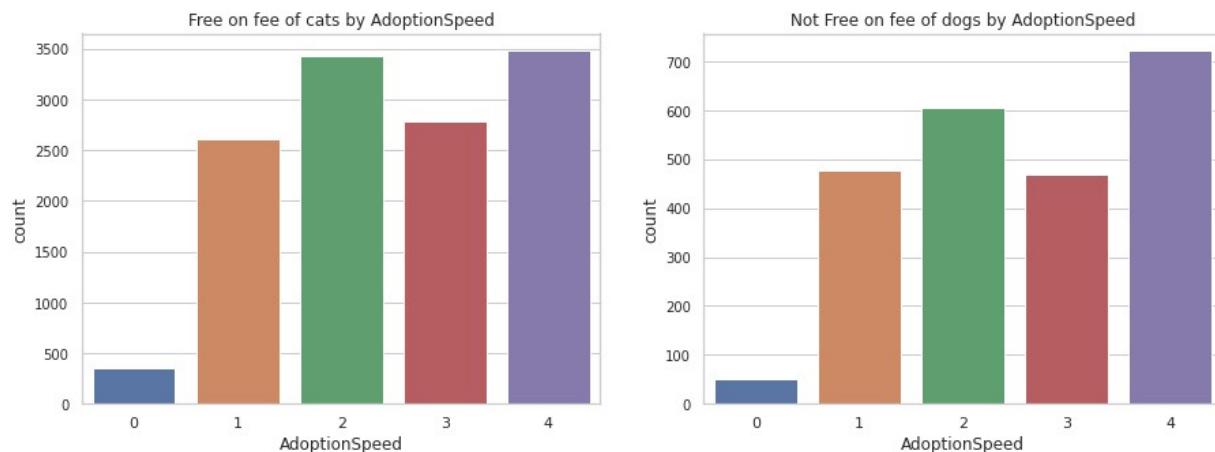
### \* Nhận xét:

- Hầu hết các con vật được nhận về trung tâm có số lượng 1 con/lượt cứu hộ, chiếm 86.6%; số lượng nhóm con vật được cứu hộ từ 2 đến 5 con/lượt chiếm 10.6%; số lượng các nhóm con vật được cứu hộ có trên 5 con/lượt chiếm 2.8%.
- Ở nhóm các con vật số lượng 1 con/lượt cứu hộ có thời gian nhận nuôi nhanh hơn đáng kể so với nhóm các con vật có từ 2 con trở lên.

### 6.2.9. Biển phí (Fee):



Hình 6.15. Biểu đồ bít phét phân phối phí nhận nuôi

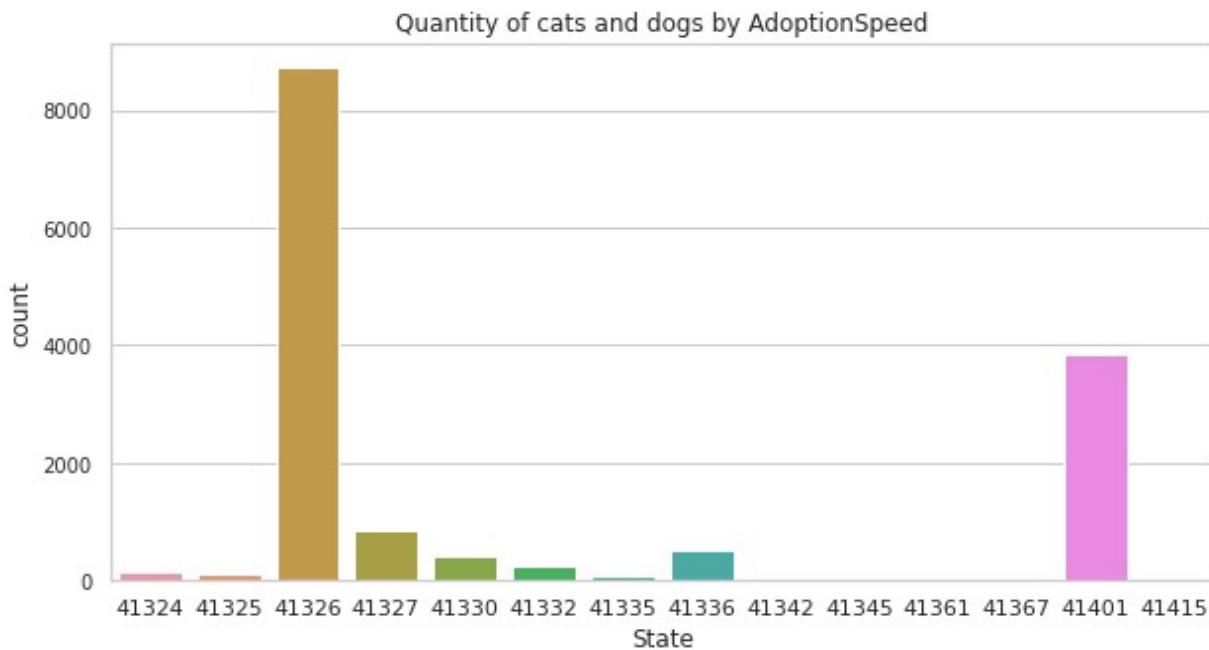


Hình 6.16. Biểu đồ bít phết phân phối phí nhận nuôi theo thời gian được nhận nuôi

### \* Nhận xét:

- Đa số các con vật khi nhận nuôi đều không phải mất phí nhận nuôi.
- Các con vật không mất phí nhận nuôi có thời gian được nhận nuôi thường từ dưới 2 tháng trong khi các con vật mất phí nhận nuôi thời gian được nhận nuôi thường trên 3 tháng.

### 6.2.10. Vùng – địa điểm nhận nuôi (State)

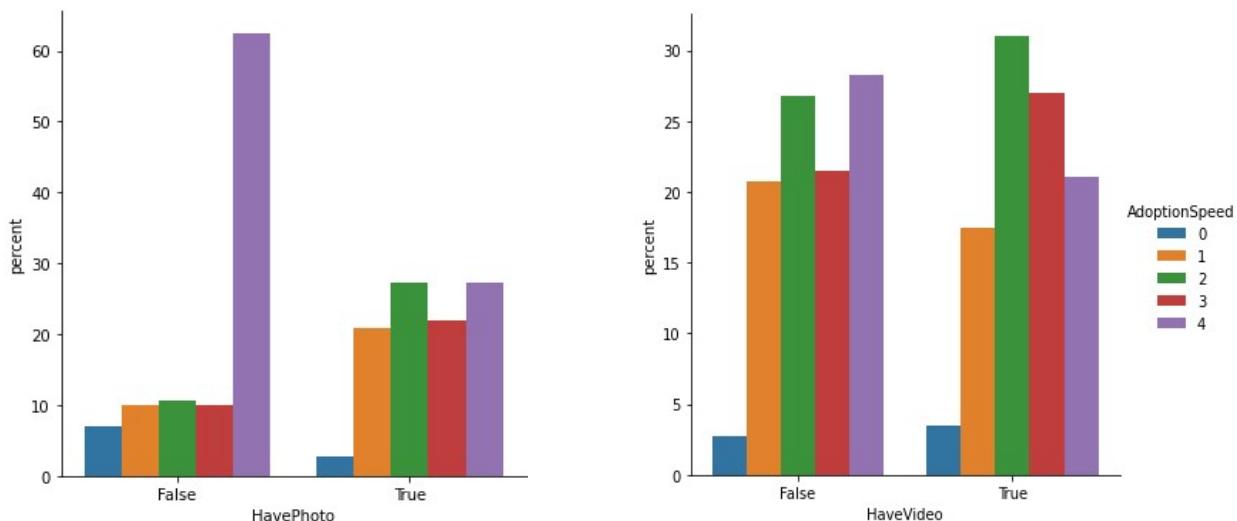


Hình 6.17. Biểu đồ biểu diễn vùng nhận nuôi con vật

\* **Nhận xét:** Đã số các con vật phân bố ở vùng có mã hiệu 41326 và 41401, các vùng khác có số lượng con vật rải rác.

### 6.2.11. Các thông số về hình ảnh của con vật (VideoAmt, PhotoAmt)

Việc đánh giá tốc độ con vật được nhận nuôi dựa vào dữ liệu hình ảnh sẽ được phân tích riêng bằng các mạng học sâu xử lý dữ liệu ảnh, ở đây ta chỉ xét tới việc con vật đó có ảnh, video đi kèm hay không.



Hình 6.18. Biểu đồ biểu diễn thông số về hình ảnh con vật

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

\* **Nhận xét:** Đa số các con vật đều có hình ảnh kèm theo, trong khi rất ít các con vật có video kèm theo.

- Các con vật không có hình ảnh và video kèm theo thường được nhận nuôi sau 100 ngày.

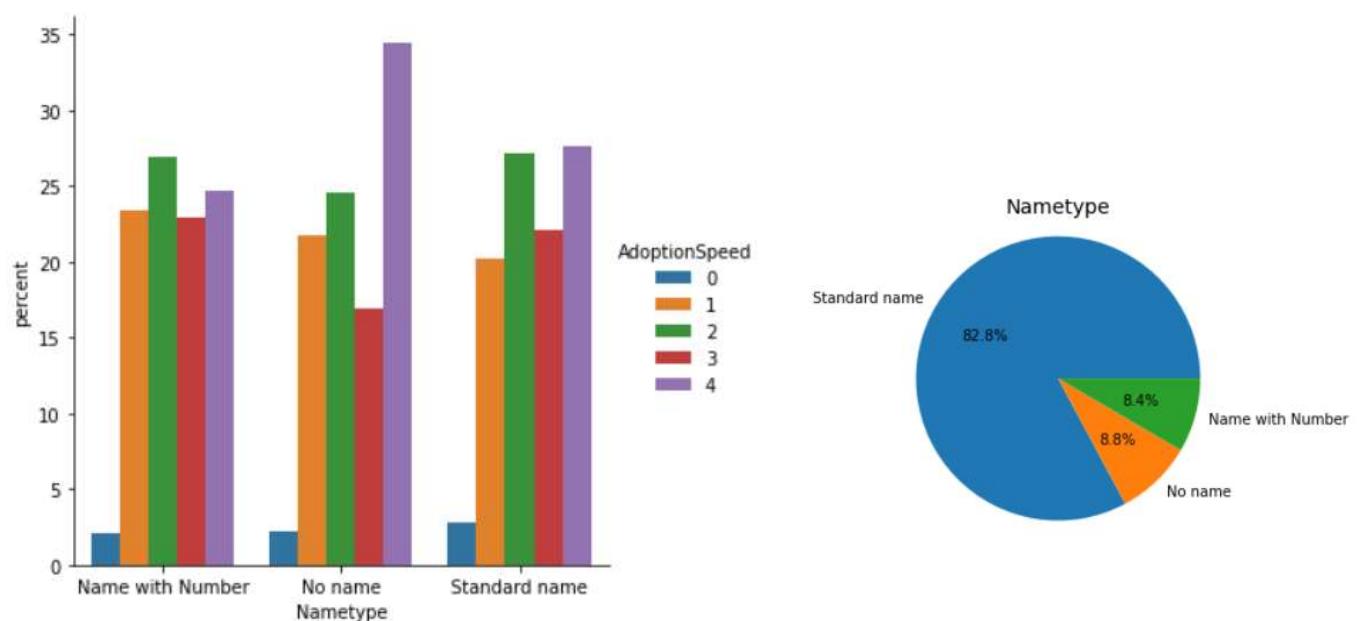
### **6.2.12. Biến Name**

Trường tên con vật thuộc kiểu string chứa nhiều các dữ liệu khác nhau. Có những con vật được đặt tên, có những con vật không có tên, có những con vật được đánh theo số thứ tự... Do vậy, để có thể thuận tiện cho quá trình mô hình hóa, ta có thể chuyển biến tên con vật vào 3 lớp:

0 - Các con vật không có tên,

1 - Các con vật được đánh tên theo số thứ tự hoặc số lượng,

2 - Các con vật đã được đặt tên

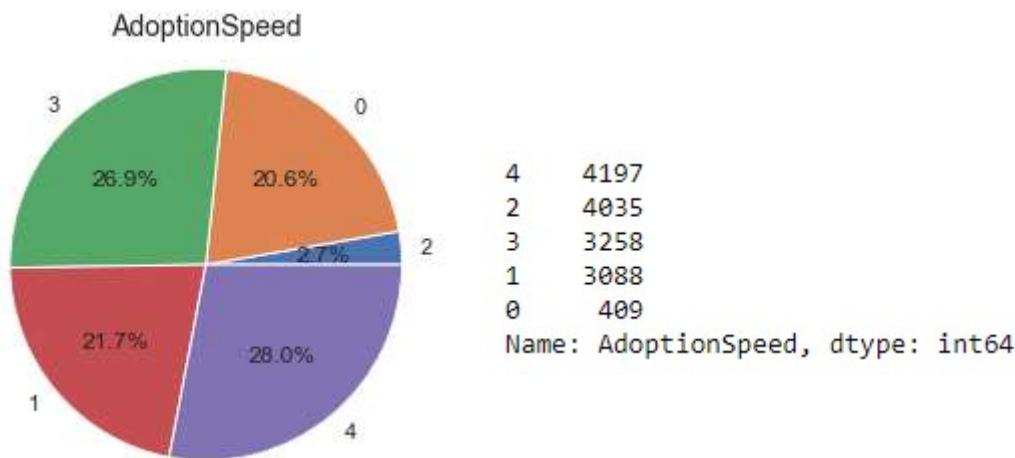


Hình 6.19. Biểu đồ kiểu đặt tên của con vật

\* **Nhận xét:** Đa số các con vật đều được đặt tên, số ít các con vật không được đặt tên hoặc đánh tên theo số khi nhận về.

- Các con vật không được đặt tên thường có thời gian nhận nuôi sau 100 ngày

### 6.2.13. Biến tốc độ nhận nuôi (AdoptionSpeed)



Hình 6.20. Biểu đồ thời gian được nhận nuôi

\* **Nhận xét:** Các con vật có xu hướng được nhận nuôi rất muộn từ khi được nhận về. Có tới 28% các con vật được nhận nuôi sau 100 ngày. Chỉ có 45% các con vật được nhận nuôi trong khoảng thời gian dưới 1 tháng. Đặc biệt có 20.6% các con vật được nhận nuôi ngay sau khi được nhận về

## 6.3. Trích chọn đặc trưng cho mô hình (Feature engineering)

### 6.3.1. Các đặc trưng đã qua xử lý:

Bộ dữ liệu ban đầu có khá ít feature do vậy ta có thể thêm các đặc trưng khác của con vật từ các đặc trưng đã có mà không làm thay đổi đặc tính của con vật. Việc thêm các đặc trưng khác của con vật từ các đặc trưng đã có giúp bộ dữ liệu mới mô tả chi tiết hơn về con vật giúp cải thiện hiệu năng của mô hình.

TÊN TRƯỜNG	MÔ TẢ
PetID	Loại bỏ
AdoptionSpeed	Dữ liệu dạng số từ rác (0, 1, 2, 3, 4)
Type	Dữ liệu dạng số rác (1, 2)
Name	Loại bỏ
NameType	Dữ liệu dạng số rác (0, 1, 2)
Age	Loại bỏ
Age(year)	Dữ liệu dạng số rác (1, 2, 3, 4, 5, 6)
Breed1	Dữ liệu dạng số rác (0..300)

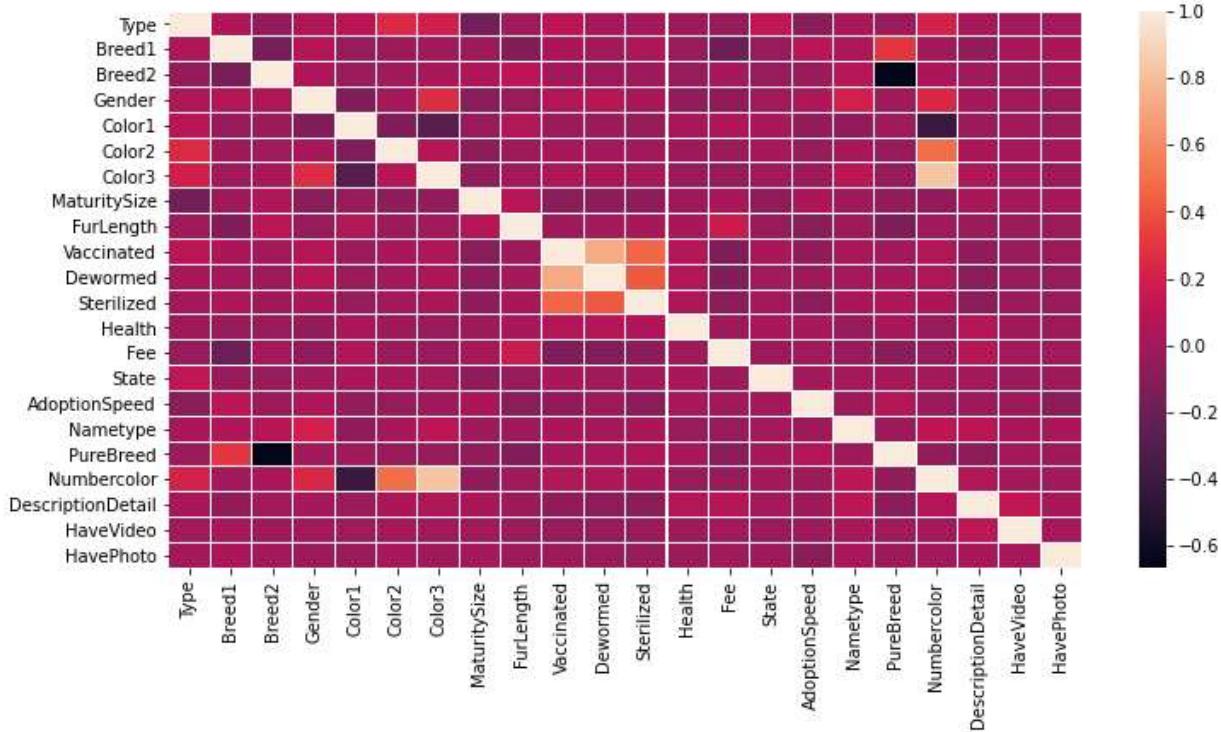
*Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng*

TÊN TRƯỜNG	MÔ TẢ
Breed2	Dữ liệu dạng số rời rạc (0..300)
PureBreed	Dữ liệu dạng số rời rạc (0, 1)
Gender	Dữ liệu dạng số rời rạc (1, 2, 3)
Color1	Dữ liệu dạng số rời rạc (0, 1, 2, 3, 4, 5, 6, 7, 8)
Color2	Dữ liệu dạng số rời rạc (0, 1, 2, 3, 4, 5, 6, 7, 8)
Color3	Dữ liệu dạng số rời rạc (0, 1, 2, 3, 4, 5, 6, 7, 8)
MaturitySize	Dữ liệu dạng số rời rạc (0, 1, 2, 3, 4)
FurLength	Dữ liệu dạng số rời rạc (0, 1, 2, 3)
Vaccinated	Dữ liệu dạng số rời rạc (0, 1, 2)
Dewormed	Dữ liệu dạng số rời rạc (0, 1, 2, 3)
Sterilized	Dữ liệu dạng số rời rạc (0, 1, 2, 3)
Health	Dữ liệu dạng số rời rạc (0, 1, 2, 3)
Quantity	Loại bỏ
QuantityRank	Dữ liệu dạng số rời rạc (0, 1, 2)
Fee	Dữ liệu dạng số rời rạc (0, 1)
State	Dữ liệu rời rạc
RescuerID	Loại bỏ
VideoAmt	Loại bỏ
HavaVideo	Dữ liệu dạng số rời rạc (0, 1)
PhotoAmt	Loại bỏ
HavePhoto	Dữ liệu dạng số rời rạc (0, 1)
Description	Loại bỏ

Bộ dữ liệu còn lại gồm 23 biến.

### 6.3.2. Phân tích tương quan giữa các biến giải thích với biến phụ thuộc:

#### 6.3.2.1 Phân tích tương quan các biến



Hình 6.21. Biểu đồ hệ số tương quan giữa các biến định lượng

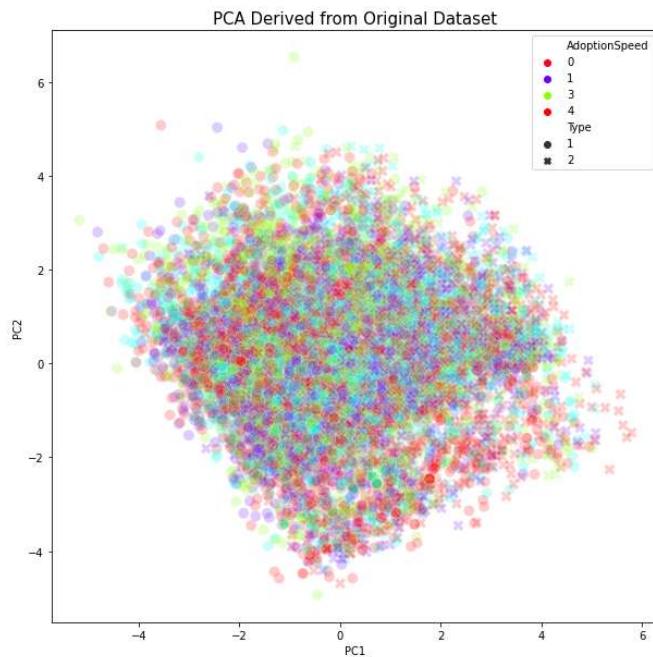
```
Breed1           0.107436
PureBreed        0.076190
Gender           0.057595
MaturitySize     0.045736
Health            0.029327
State             0.013450
Fee               0.000769
Color3            -0.006700
Nametype          -0.008717
DescriptionDetail -0.010250
HaveVideo         -0.011717
Dewormed          -0.013247
Numbercolor       -0.015752
Breed2            -0.018857
Color2            -0.038688
Color1            -0.044326
Vaccinated        -0.059200
HavePhoto         -0.076694
Sterilized         -0.083335
FurLength          -0.090835
Type              -0.091599
Name: AdoptionSpeed, dtype: float64
```

Hình 6.22. Tương quan biến AdoptionSpeed với các biến còn lại.

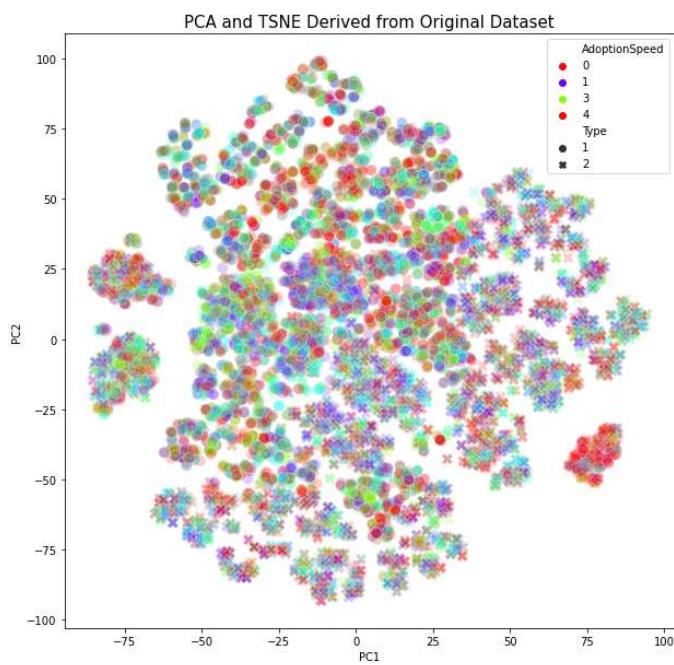
**Nhận xét:** Ta thấy sự tương quan giữa biến Adoption với các biến còn lại trong mô hình không có sự khác biệt rõ ràng.

### 6.3.2.2 Biểu diễn các biến giải thích và biến phụ thuộc

\* Phân tích thành phần chính (PCA)



Hình 6.23. Biểu đồ phân bố các biến bằng phương pháp PCA

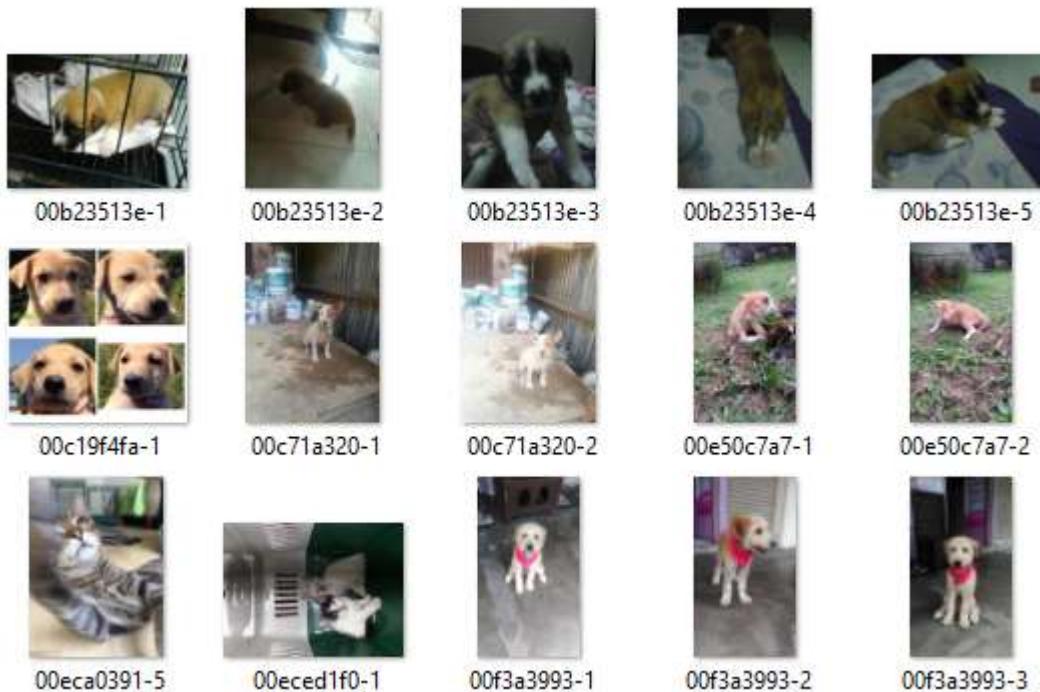


Hình 6.24. Biểu đồ phân bố các biến bằng phương pháp PCA kết hợp TSNE

## 7. PHÂN TÍCH TRÍCH CHỌN ĐẶC TRƯNG TỪ DỮ LIỆU ẢNH

### 7.1. Giới thiệu bộ dữ liệu:

Bộ dữ liệu gồm 58.311 bức ảnh được chụp cho 14.993 con vật đã được cứu hộ về trạm giải cứu động vật



*Hình 7.1. Giới thiệu bộ dữ liệu: Hình ảnh 1 số con vật được cứu hộ*

#### Nhận xét:

- Trong bộ dữ liệu gồm các ảnh được chụp bởi nhiều góc độ khác nhau, có bức chụp ngang, có bức chụp dọc...
- Kích thước các bức ảnh có kích thước độ phân giải khác nhau.
- Mỗi con vật được cứu hộ về có thể có 1 hay nhiều bức ảnh khác nhau được lưu trữ trong cơ sở dữ liệu.
- Trong bức ảnh có thể bao gồm 1 hoặc nhiều hình ảnh của con vật.

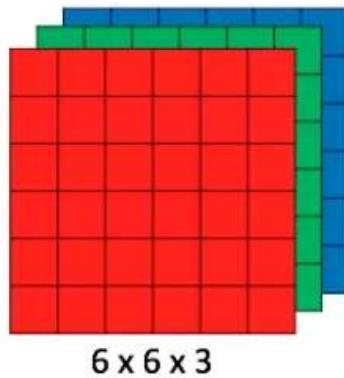
### 7.2. Tổng quan về deep learning – mạng tích chập CNN

Máy tính ngày nay không thể tự động đọc hiểu để phân loại các hình ảnh. Nó chỉ đơn giản là đọc các điểm hình ảnh và đưa các dữ liệu về màu sắc các điểm ảnh khác nhau để có thể tạo nên một bức ảnh hoàn chỉnh. Trong Deep Learning chúng ta có các mô hình Neural tích chập (CNN) để nhận dạng và phân loại các hình ảnh.

CNN phân loại hình ảnh bằng cách lấy 1 hình ảnh đầu vào vào xử lý phân loại nó theo hạng mục nhất định. Máy tính coi ảnh đầu vào như 1 mảng pixel và nó phụ thuộc vào độ phân giải

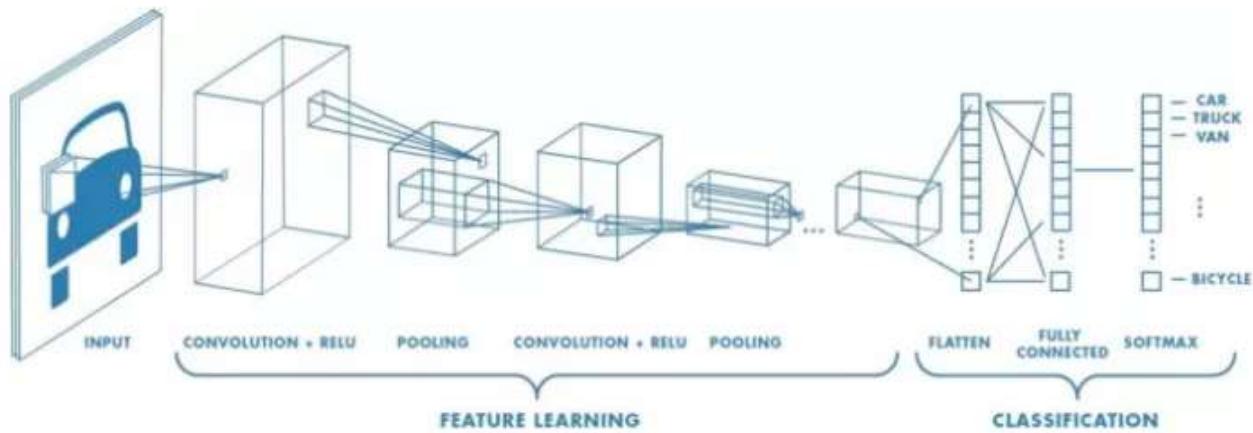
## Dự đoán khả năng được nhận nuôi của động vật từ trạm cùu hộ với dữ liệu đa dạng

của hình ảnh. Dựa trên độ phân giải của hình ảnh, máy tính sẽ thấy  $H \times W \times D$  ( $H$ : Chiều cao,  $W$ : Chiều rộng,  $D$ : Độ dày). Ví dụ: Hình ảnh là một mảng ma trận RGB  $6 \times 6 \times 3$ .



Hình 7.2. Hình ảnh 1 một mảng ma trận RGB  $6 \times 6 \times 3$

Về kỹ thuật, mô hình CNN để training và kiểm tra, mỗi hình ảnh đầu vào sẽ chuyển nó qua 1 loạt các lớp tích chập với các bộ lọc (Kernels), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1. Hình dưới đây là toàn bộ luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.



Hình 7.3. Luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị

- Các lớp tích chập (Convolution);
  - Tích chập là lớp đầu tiên để trích xuất các thuộc tính từ hình ảnh đầu vào. Sau các lớp tích chập, mỗi quan hệ giữa các pixel vẫn được duy trì bằng cách thực hiện phép toán tuyến tính có 2 đầu vào là ma trận hình ảnh và một bộ lọc kernel.
  - Ví dụ xét 1 ma trận  $5 \times 5$  có giá trị pixel là 0 và 1. Ma trận kernel  $3 \times 3$  như hình dưới đây:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

\*

1	0	1
0	1	0
1	0	1

5 x 5 – Image Matrix

3 x 3 – Filter Matrix

Hình 7.4. Các lớp tích chập 1

- Sau phép tích chập ra được một ma trận 3x3 các thuộc tính của ma trận 5x5 ban đầu:

1	1	1	0	0
0	1	1	1	0
0	0	1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>
0	0	1 <sub>x0</sub>	1 <sub>x1</sub>	0 <sub>x0</sub>
0	1	1 <sub>x1</sub>	0 <sub>x0</sub>	0 <sub>x1</sub>

Image

4	3	4
2	4	3
2	3	4

Convolved  
Feature

Hình 7.5. Các lớp tích chập 2

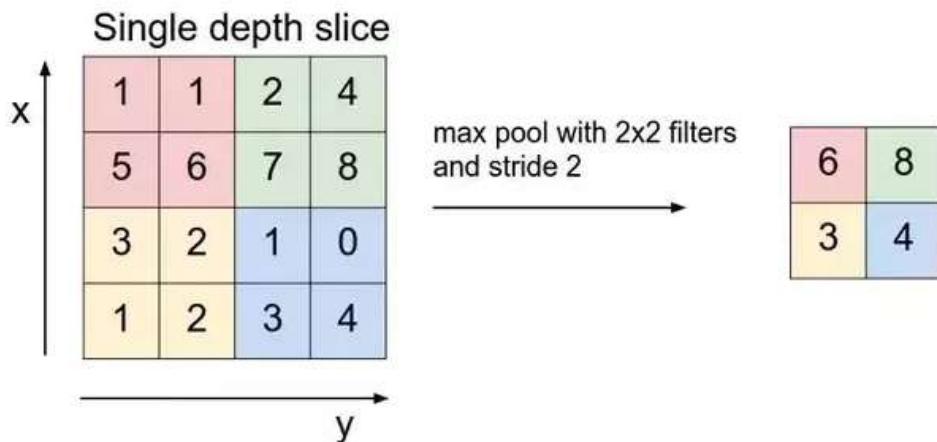
- Sự kết hợp của 1 hình ảnh với các bộ lọc khác nhau có thể có các kết quả khác nhau nhằm phát hiện ra các thuộc tính ẩn của bức ảnh.

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Hình 7.6. Các lớp tích chập 3

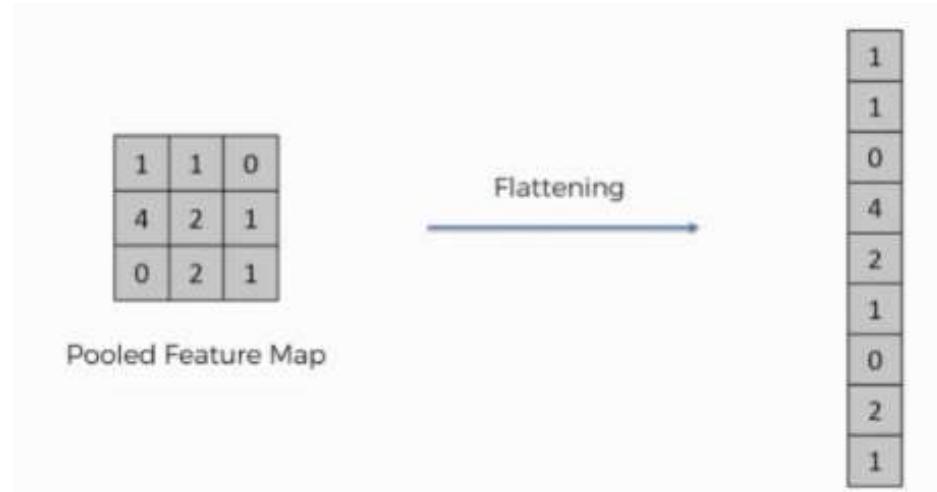
- Lớp gộp (Pooling layer):

- Lớp pooling sẽ giảm bớt số lượng tham số khi hình ảnh quá lớn. Không gian pooling còn được gọi là lấy mẫu con hặc lấy mẫu xuống làm giảm kích thước của mỗi ma trận thuộc tính nhưng vẫn giữ được các thông tin quan trọng. Các pooling có nhiều loại khác nhau: Max pooling, Average pooling, Sum pooling.
- Ví dụ max pooling là lấy giá trị lớn nhất từ ma trận đối tượng:



Hình 7.7. Lớp gộp

- Lớp Fully connected:
  - Sau khi hình ảnh được truyền qua nhiều lớp Convolutional layer và pooling layer thì máy tính đã học được tương đối các đặc điểm của hình ảnh thì ma trận tính cuối cùng có kích thước  $H \times W \times D$  sẽ được chuyển về thành 1 vector kích thước  $(H \times W \times D)$



Hình 7.8. Lớp fully connected

Sau đó dữ liệu sẽ được xử lý qua lớp fully connected để kết hợp các

### 7.3. Khó khăn thách thức

Để có thể tạo được một mạng học sâu có thể phân loại được các con vật theo thời gian được nhận nuôi của chúng. Ta buộc phải có các dữ liệu đã được gán nhãn trước để có thể cho máy tính đọc hiểu được.

Các hình ảnh đầu vào của mạng CNN phải được chuẩn hóa về cùng 1 kích thước nên việc bắt buộc ta phải xử lý ảnh đầu vào để có thể đưa vào mô hình.

## 7.4. Xử lý dữ liệu đầu vào

### 7.4.1. Chuẩn hóa kích thước hình ảnh:

Trong bộ dữ liệu ta có 14.993 id con vật trong khi có tới 58.311 bước ảnh. Vậy nên ta cần phải loại bỏ các ảnh thừa của con vật đi trước khi xử lý các bước tiếp theo:

```
pet_ids = train_df['PetID'].values

%%time
X = []
Y = []
petid = []
for pet_id in tqdm(pet_ids):
    try:
        im = load_image(config.root + "train_images/" + pet_id + '-1.jpg')
        X.append(im)
        ads = train_df[train_df['PetID'] == pet_id]['AdoptionSpeed'].values[0]
        ids = train_df[train_df['PetID'] == pet_id]['PetID'].values[0]
        Y.append(ads)
        petid.append(ids)
    except:
        pass
X = np.asarray(X)
```

100% | 14993/14993 [19:45<00:00, 12.64it/s]

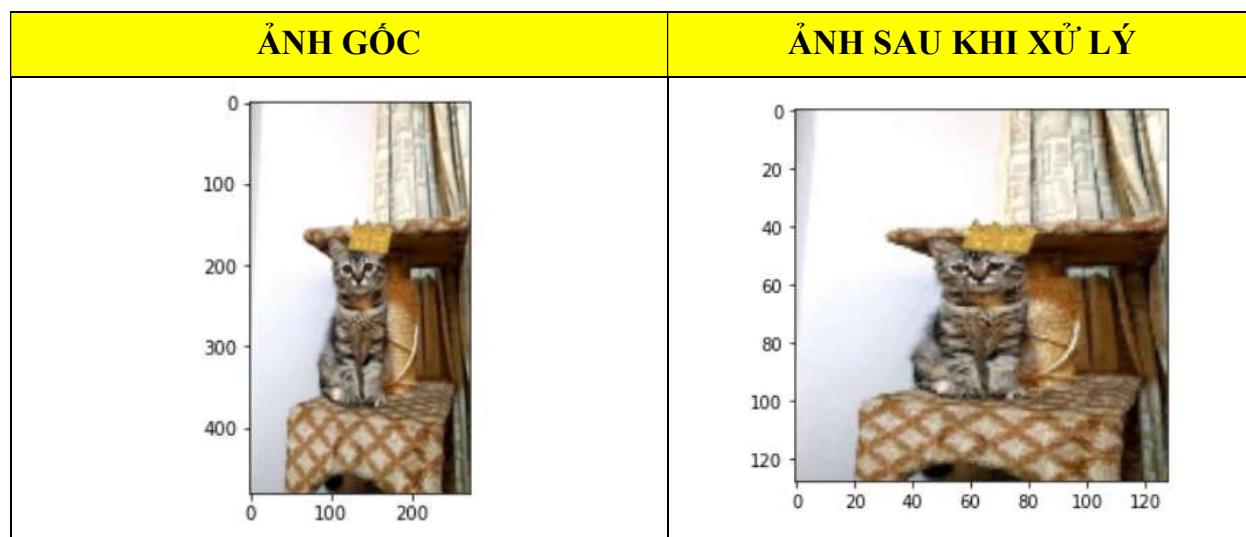
Hình 7.9. Chuẩn hóa các kích thước ảnh

Để làm điều này ta sẽ lọc tất cả các id của con vật, sau đó lựa chọn trong tập các ảnh các ảnh đầu tiên của id con vật đó.

### 7.4.2. Chuẩn hóa kích thước hình ảnh:

Để có thể đưa dữ liệu hình ảnh vào mạng học sâu. Các hình ảnh cần phải có cùng 1 kích thước đầu vào. Do đó việc tiên quyết ta cần phải đưa các hình ảnh về cùng một kích thước ban đầu. Có nhiều phương pháp để xử lý kích thước ảnh có thể sử dụng, tuy nhiên mỗi phương pháp có một ưu nhược điểm khác nhau:

- Phương pháp resize bức ảnh về cùng 1 kích thước

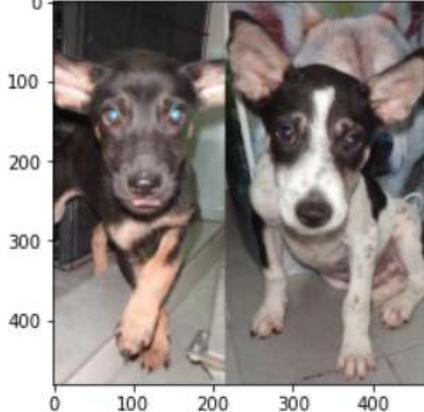
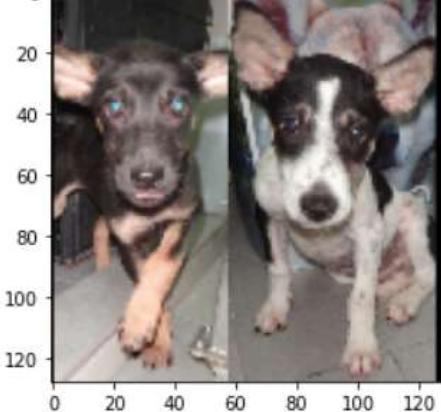


## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

ẢNH GỐC	ẢNH SAU KHI XỬ LÝ
	

**Nhận xét:** Hình ảnh sau khi resize bị biến dạng, thay đổi tỉ lệ khung hình, do đó có thể ảnh hưởng tới hiệu năng của mô hình.

- Resize kết hợp padding:

ẢNH GỐC	ẢNH SAU KHI XỬ LÝ
	
	

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

**Nhận xét:** Việc kết hợp padding để resize đã xử lý được việc hình ảnh bị biến dạng. Tuy nhiên với các bức ảnh có nhiều con vật, các vật thể khác không liên quan tới con vật vẫn được giữ nguyên trong bức ảnh.

- Crop kết hợp padding và resize:

Để có thể xử lý việc có nhiều các đối tượng không mong muốn có trong hình ảnh, ta có thể crop ảnh đi. Tuy nhiên con vật có thể nằm ở vị trí bất kì trong hình ảnh, do đó việc crop tại vị trí nào để không bị mất đi hình ảnh con vật cũng cần phải xét tới. Để giải quyết bài toán trên, ta sẽ sử dụng OpenCV để phát hiện vị trí của con chó hoặc mèo trong bức ảnh, sau đó sẽ crop vị trí đó:

Trong mô hình, nhóm sử dụng mô hình “ssd\_mobilenet\_v3” được xây dựng trên bộ dữ liệu COCO Dataset. Mô hình sẽ giúp phát hiện và khoanh vùng đối tượng là chó hoặc mèo. Với những bức ảnh có nhiều hơn 1 đối tượng con vật, nhóm sẽ lựa chọn đối tượng có chỉ số accuracy cao nhất để trích xuất:



Figure 1



Figure 2

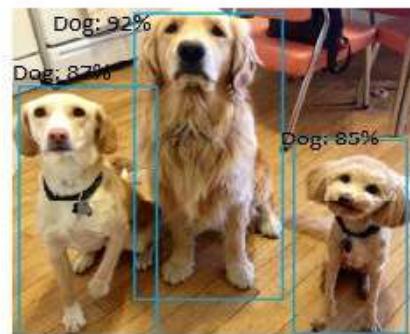


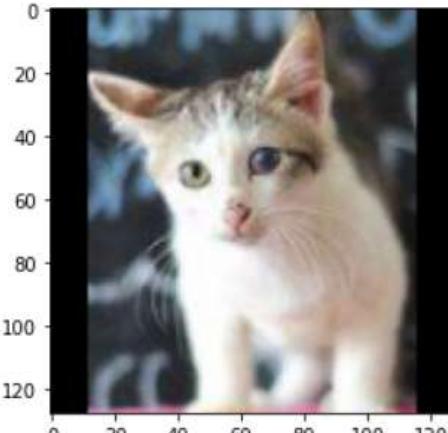
Figure 3

Hình 7.10. Mô hình ssd\_mobilenet\_v3 giúp phát hiện và khoanh vùng đối tượng chó mèo

Kết quả ta thu được:

ẢNH GỐC	ẢNH SAU KHI XỬ LÝ

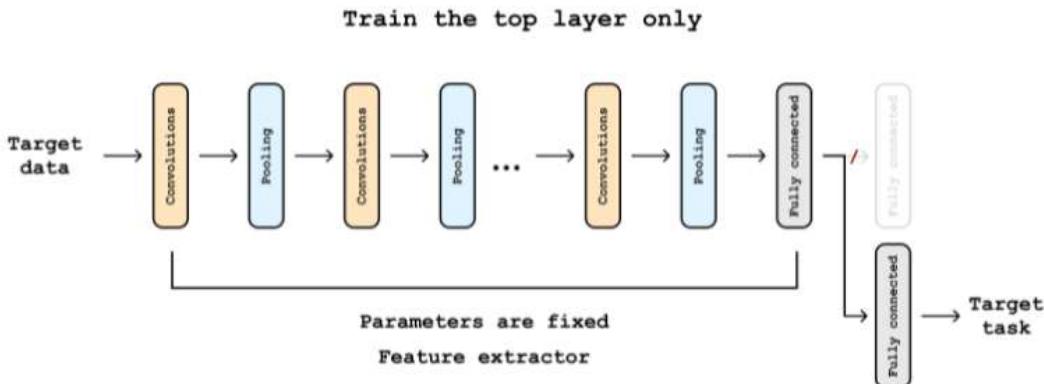
## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

ẢNH GÓC	ẢNH SAU KHI XỬ LÝ
	
	

Kết quả thu được là các hình ảnh đã được crop lấy đối tượng và được resize về cùng kích thước 128x128x3

### 7.5. Xây dựng mô hình

Để xây dựng mô hình cho bài toán, nhóm sử dụng mạng học sâu Resnet50 sau đó nhóm bỏ đi lớp FC cuối cùng và thêm 1 lớp Flatten phía sau. Sau đó dùng hàm activation = ‘relu’ để đưa ra kết quả là 1 Dense gồm 5 class là thời gian được nhận nuôi của từng con vật



Hình 7.11. Xây dựng mô hình

Sau khi dựng mô hình nhóm tiến hành chia tập dữ liệu ra thành tập training và tập test tiến hành huấn luyện với mô hình. Trong bài toán nhóm sử dụng hàm mất mát Kappa\_cohen để compile:

```
opt = Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=None, decay=0.0, amsgrad=False)
model.compile(loss = tfa.losses.KappaLoss(num_classes = nb_classes),
              optimizer=opt,
              metrics=['accuracy'])
```

```
Epoch 40/50
277/277 - 253s - loss: 0.1456 - accuracy: 0.3551 - val_loss: 0.1494 - val_accuracy: 0.3452
Epoch 41/50
277/277 - 254s - loss: 0.1455 - accuracy: 0.3576 - val_loss: 0.1493 - val_accuracy: 0.3442
Epoch 42/50
277/277 - 252s - loss: 0.1456 - accuracy: 0.3568 - val_loss: 0.1495 - val_accuracy: 0.3269
Epoch 43/50
277/277 - 253s - loss: 0.1456 - accuracy: 0.3571 - val_loss: 0.1493 - val_accuracy: 0.3401
Epoch 44/50
277/277 - 253s - loss: 0.1453 - accuracy: 0.3631 - val_loss: 0.1515 - val_accuracy: 0.3228
Epoch 45/50
277/277 - 242s - loss: 0.1453 - accuracy: 0.3567 - val_loss: 0.1505 - val_accuracy: 0.3401
Epoch 46/50
277/277 - 230s - loss: 0.1452 - accuracy: 0.3612 - val_loss: 0.1495 - val_accuracy: 0.3523
Epoch 47/50
277/277 - 230s - loss: 0.1451 - accuracy: 0.3597 - val_loss: 0.1497 - val_accuracy: 0.3473
Epoch 48/50
277/277 - 229s - loss: 0.1452 - accuracy: 0.3640 - val_loss: 0.1498 - val_accuracy: 0.3381
Epoch 49/50
277/277 - 229s - loss: 0.1449 - accuracy: 0.3647 - val_loss: 0.1505 - val_accuracy: 0.3432
Epoch 50/50
277/277 - 231s - loss: 0.1450 - accuracy: 0.3603 - val_loss: 0.1496 - val_accuracy: 0.3442
Wall time: 3h 22min 24s
```

Hình 7.12. Huấn luyện tập train và tập test với mô hình đã dựng

Kết quả trên tập test đạt độ chính xác 31.49%

```
scores = model.evaluate(X_tst, Y_tst, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

Accuracy: 31.49%

Hình 7.13. Kết quả trên tập test

## 8. PHÂN TÍCH DỮ LIỆU DẠNG VĂN BẢN

### 8.1. Tổng quan về xử lý ngôn ngữ tự nhiên (NLP)

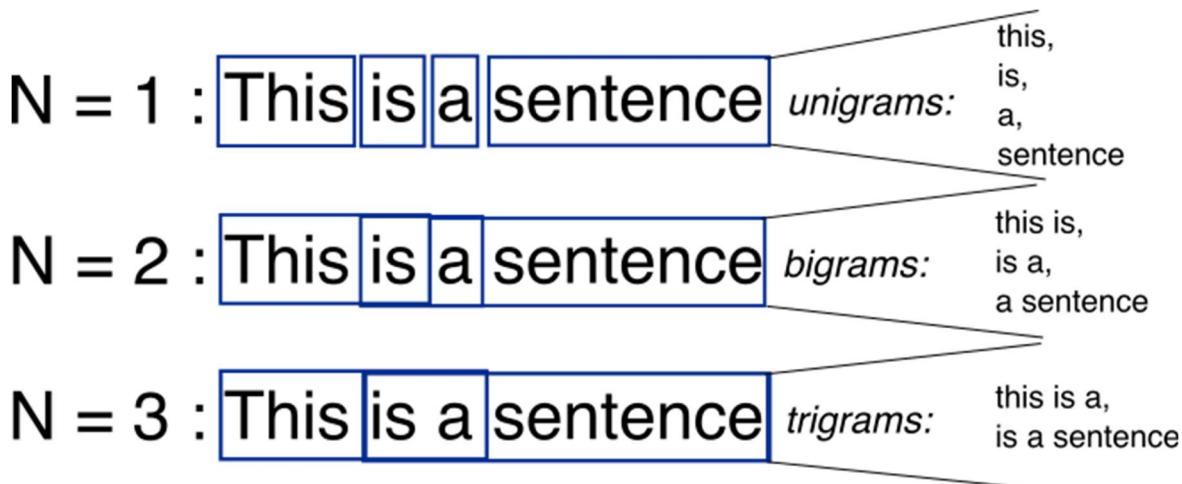
Xử lý ngôn ngữ tự nhiên (*natural language processing* - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất.

Ngôn ngữ tự nhiên là những ngôn ngữ được con người sử dụng trong các giao tiếp hàng ngày. Mặc dù con người có thể dễ dàng hiểu được và học các ngôn ngữ tự nhiên nhưng việc làm cho máy hiểu được ngôn ngữ tự nhiên không phải là chuyện dễ dàng (cấu trúc ngữ pháp, ngữ cảnh...)

Các phương pháp xử lý ngôn ngữ tự nhiên dựa trên thống kê với mục đích cho máy tính có thể “học” nhờ vào việc thống kê các từ và cụm từ (N-gram) có trong văn bản, từ đó xây dựng mô hình ngôn ngữ.

#### 8.1.1. N-gram

- N-gram được hiểu đơn giản là tần suất xuất hiện của n kí tự (từ) liên tiếp xuất hiện trong dữ liệu.
- Một số mô hình n-gram phổ biến:
  - Unigram, mô hình với  $n=1$ , tức là ta sẽ tính tần suất xuất hiện của một kí tự (từ), như: "k", "a", ...
  - Bigram với  $n=2$ , là mô hình được sử dụng nhiều trong việc phân tích các hình thái cho ngôn ngữ.
  - Trigram với  $n=3$ , với  $n$  càng lớn thì độ chính xác càng cao tuy nhiên đi kèm với đó thì độ phức tạp cũng lớn hơn.



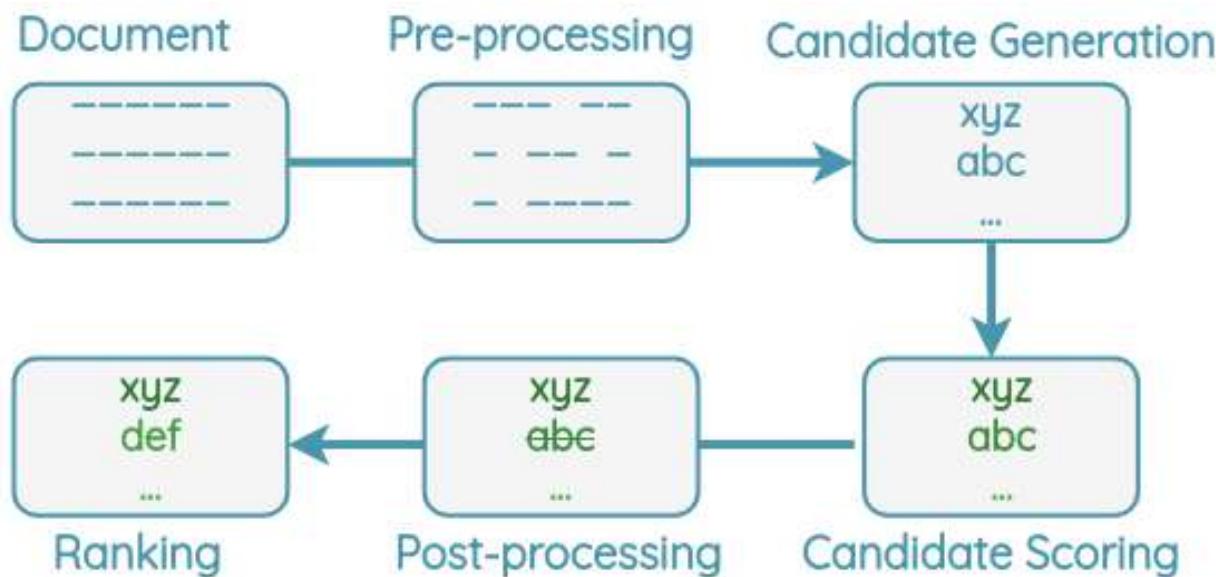
Hình 8.1. N-gram

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cừu hộ với dữ liệu đa dạng

Chúng ta có thể sinh ra các từ khóa ứng viên bằng cách tách từ theo khoảng trắng. Khi đó ta sẽ được các từ đơn (1-gram). Ngoài ra, ta có thể ghép 2 từ liên tiếp để tạo ra cụm từ 2-gram, 3 từ liên tiếp để tạo ra 3-gram,... và đó sẽ là các ứng cử viên từ khóa.

### **8.1.2. Trích rút từ khóa trong văn bản**

Trích rút từ khóa là một trong những phương pháp đơn giản nhất giúp cho việc phân tích & khai thác các giá trị từ dữ liệu văn bản. Bài toán trích rút từ khóa (tiếng anh: Keyword Extraction hoặc Keyphrase Extraction) là quá trình tự động trích rút ra các từ khóa/ thuật ngữ là các từ/cụm từ tiêu biểu, đại diện cho văn bản.



Hình 8.2. Trích rút từ khóa trong văn bản

- Văn bản (Document) cần trích rút từ khóa sẽ đi qua bước tiền xử lý (Pre-processing) để chuẩn hóa cũng như loại bỏ các thông tin gây nhiễu hoặc có rất ít giá trị cho việc trích rút từ khóa. Chẳng hạn như xóa bỏ các mã HTML, loại bỏ các stopword, dấu câu,...
- Từ văn bản đã tiền xử lý, ta sẽ tìm cách lấy ra các ứng viên (Candidate) có khả năng là từ khóa (từ hoặc cụm từ) đại diện cho văn bản.
- Mỗi từ khóa ứng viên sẽ được đánh giá bằng thuật toán và được gán một điểm số nhất định. Các từ khóa có điểm số cao nhất sẽ được lựa chọn.
- Các từ khóa tiềm năng sẽ qua bước hậu xử lý (Post-processing). Bước này sẽ giúp ta sàng lọc một lần nữa, chẳng hạn như loại bỏ các từ khóa gần giống nhau, có cùng ý nghĩa.
- Cuối cùng, ta sẽ có bảng xếp hạng các từ khóa ứng viên đó và chúng ta sẽ lấy ra top N từ khóa làm kết quả cuối cùng.

### **8.1.3. Tiền xử lý văn bản và Nhận dạng cụm từ Úng viên**

Mục tiêu của tiền xử lý văn bản là chuẩn hóa và làm sạch tài liệu. Quy trình xử lý thông tin nhằm mục đích chuyển đổi văn bản thành một định dạng thống nhất cho phép xử lý hiệu quả hơn. Các kỹ thuật chuẩn hóa phổ biến bao gồm chuyển đổi các ký tự thành chữ thường, mã hóa, tách câu, đưa về dạng chuẩn, góc từ. Quá trình làm sạch xác định và tùy chọn loại bỏ các ký tự hoặc từ mang ít hoặc không có ý nghĩa ngữ nghĩa, chẳng hạn như dấu chấm câu, từ dừng (stop-word), ký hiệu hoặc phương trình toán học. Các kỹ thuật được sử dụng trong làm sạch văn bản có thể phụ thuộc vào ngữ liệu, thường yêu cầu áp dụng phương pháp phỏng đoán để loại bỏ thông tin không cần thiết hoặc không liên quan.

### **8.1.4. Nhận dạng cụm từ Úng viên**

Cụm từ khóa không chỉ bao gồm các từ đơn mà còn bao gồm các cụm từ nhiều từ, và do đó quy trình nhận dạng cụm từ ứng viên nhận dạng các cụm từ có thể chấp nhận được về mặt cú pháp từ các tài liệu, được coi là ứng viên của cụm từ khóa. Hai cách tiếp cận phổ biến là Cụm từ dưới dạng phân đoạn văn bản và Nhận dạng cụm từ với các mẫu Part-of-Speech.

### **8.1.5. Các đặc điểm cụm từ phổ biến**

Sau khi xác định các cụm từ ứng viên, quá trình tiếp theo là chọn các đặc điểm của cụm từ dựa trên quan sát của tập dữ liệu. Các đặc trưng đại diện cho các ký tự riêng biệt của một cụm từ, giúp phân biệt chính nó với các cụm từ khác. Nói chung, một cụm từ có hai loại đặc trưng là độc lập và quan hệ. Các đặc trưng độc lập liên quan đến thông tin về chính một cụm từ, chẳng hạn như tần suất của nó, cấu trúc ngôn ngữ hoặc các vị trí xuất hiện của nó trong một tài liệu. Các đặc trưng quan hệ nắm bắt thông tin quan hệ của một cụm từ với những cụm từ khác, chẳng hạn như quan hệ đồng xuất hiện và quan hệ ngữ nghĩa.

#### **8.1.5.1. Các đặc trưng độc lập**

**1. Tần suất** là nguồn chính để xác định tầm quan trọng của các cụm từ. Tuy nhiên, trong nhiều trường hợp, thống kê tần suất thô không phản ánh chính xác tầm quan trọng của các cụm từ. Một cụm từ có tần suất cao khác biệt có thể không phải là dấu hiệu phân biệt tốt để trở thành cụm từ khóa nếu nó phân bố đồng đều trong một ngữ liệu.

**2. Các đặc trưng ngôn ngữ** thu được từ phân tích ngôn ngữ, bao gồm gắn thẻ từ loại (POS – tagging), phân tích cú pháp câu, cấu trúc cú pháp và hậu tố phụ thuộc, phân tích hình thái hậu tố hoặc tiền tố và phân tích từ vựng. Chúng chủ yếu được sử dụng trong các phương pháp tiếp cận học máy có giám sát. Tuy nhiên, với thực tế là độ dài trung bình của các cụm từ rất ngắn, các cụm từ có cấu trúc cú pháp tương đối đơn giản, điều này có thể mang lại ít lợi ích hơn cho hiệu suất tổng thể của một hệ thống.

**3. Các đặc trưng cấu trúc** mã hóa cách một cụm từ xuất hiện trong tài liệu, chẳng hạn như vị trí tương đối của lần xuất hiện đầu tiên hoặc cuối cùng, cho dù cụm từ có xuất hiện trong phần tóm tắt của tài liệu hay không. Nói chung, trong các tài liệu có cấu trúc tốt,

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

chẳng hạn như các ấn phẩm khoa học, các cụm từ khóa có nhiều khả năng xuất hiện trong phần tóm tắt và phần giới thiệu. Các đặc điểm cấu trúc cũng có thể đặc trưng cho định dạng của văn bản.

**4. Độ dài của một cụm từ** là số lượng từ trong một cụm từ cũng được coi là một đặc trưng hữu ích.

**5. Cụm từ ghép (phraseness)** đề cập đến khả năng một chuỗi từ tạo thành một cụm từ có nghĩa. Đặc biệt, đặc trưng cụm từ ghép rất hữu ích để xác định cụm từ khóa từ các cụm từ dài hơn hoặc ngắn hơn của nó.

### **8.1.5.2 Các đặc trưng quan hệ**

**1. Thông kê cụm từ đồng xuất hiện:** cung cấp thông tin về cách các cụm từ đồng xuất hiện với những cụm từ khác. Một cụm từ đồng xuất hiện thường xuyên với những cụm từ khác cho biết bản thân cụm từ đó có tầm quan trọng cao hơn.

**2. Quan hệ cụm từ và tài liệu** kiểm tra mức độ quan trọng của một cụm từ ứng cử viên đối với một tài liệu cụ thể đối với sự phân bố của cụm từ trong ngữ liệu. TF-IDF là thuật toán phổ biến nhất, được tính là tích của tần suất cụm từ (thuật ngữ) (TF) và tần suất tài liệu nghịch đảo (IDF). TF-IDF ấn định điểm thấp hơn cho các cụm từ được phân bổ đồng đều trên kho ngữ liệu và điểm cao hơn cho các cụm từ thường xuyên xuất hiện trong một số tài liệu cụ thể.

**3. Đặc trưng ngữ nghĩa** thể hiện mối liên hệ ngữ nghĩa giữa các cụm từ, có thể là bất kỳ mối quan hệ ngôn ngữ nào giữa hai cụm từ, chẳng hạn như sự tương tự, từ đồng nghĩa, từ trái nghĩa, từ siêu nghĩa và từ viết tắt. Sự liên quan về ngữ nghĩa có thể trực tiếp thu được từ các cơ sở tri thức ngữ nghĩa không có sẵn.

**4. Các cụm từ và đặc trưng quan hệ chủ đề** cho biết ứng viên có liên quan như thế nào đến một chủ đề cụ thể trong tài liệu.

### **8.1.6. TF-IDF**

**TF-IDF** (Term Frequency – Inverse Document Frequency) là 1 kĩ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản.

TF-IDF cũng được sử dụng để lọc những từ stopwords trong các bài toán như tóm tắt văn bản và phân loại văn bản.

$$\text{TF-IDF}(t, d, D) = \text{TF}(t,d) \times \text{IDF}(t,D)$$

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

### 8.1.6.1. TF

**TF:** Term Frequency (Tần suất xuất hiện của từ) là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, term frequency thường được chia cho độ dài văn bản (tổng số từ trong một văn bản).

$$tf(t, d) = \frac{f(t, d)}{\max\{f(w, d) : w \in d\}}$$

Trong đó:

- $tf(t, d)$ : tần suất xuất hiện của từ  $t$  trong văn bản  $d$
- $f(t, d)$ : Số lần xuất hiện của từ  $t$  trong văn bản  $d$
- $\max(\{f(w, d) : w \in d\})$ : Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản  $d$

### 8.1.6.2. IDF

**IDF:** Inverse Document Frequency (Nghịch đảo tần suất của văn bản), giúp đánh giá tầm quan trọng của một từ . Khi tính toán TF , tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng một số từ như “is”, “of” và “that” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.

$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Trong đó:

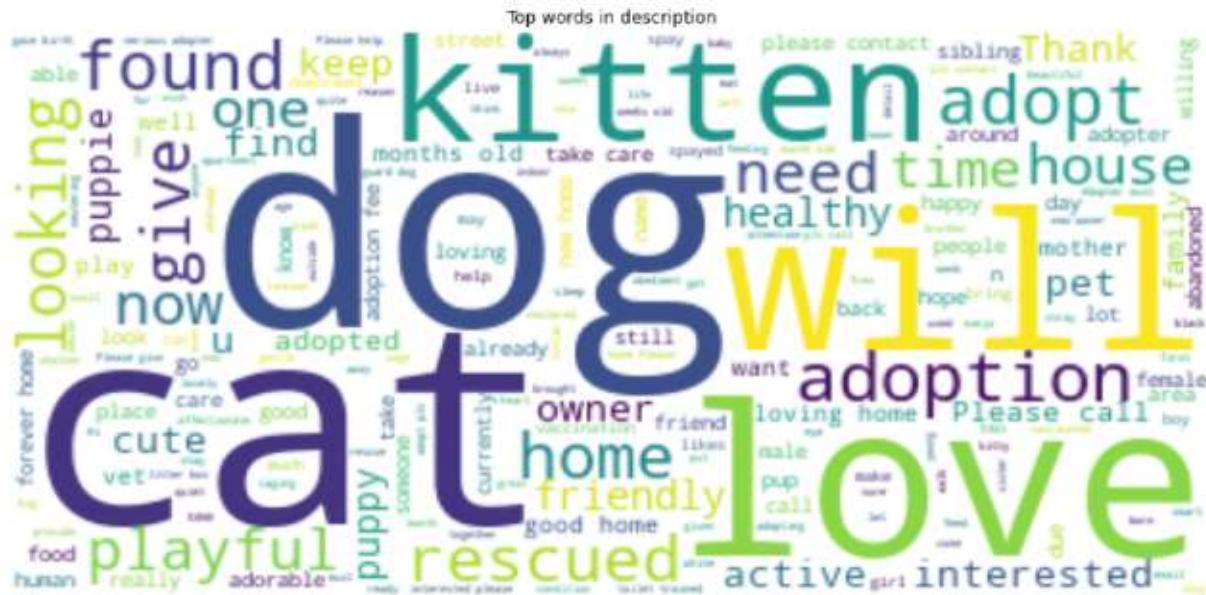
- $idf(t, D)$ : giá trị idf của từ  $t$  trong tập văn bản
- $|D|$ : Tổng số văn bản trong tập  $D$
- $|\{d \in D : t \in d\}|$ : số văn bản trong tập  $D$  có chứa từ  $t$ .

## 8.2. Phân tích trường dữ liệu Description

Trường Description là hồ sơ mô tả của con vật. Ngôn ngữ chính được sử dụng là tiếng Anh, một số bằng tiếng Malaysia hoặc Trung Quốc.

### 8.2.1. Những từ phổ biến nhất

Trong phần thông tin mô tả của con vật được cứu hộ, những từ xuất hiện phổ biến: Cat, Dog, Love, Kitten, Found Adoption, Will, Love, Playful...mô tả tình trạng và các đặc điểm của con vật như: loại, giới tính, tình trạng, nét ngoại hình nổi bật nhất (như màu lông)...



Hình 8.3. Những từ phổ biến nhất trong trường Description

Một trường dữ liệu khác cũng có dạng văn bản là Name.



Hình 8.4. Những từ phổ biến nhất trong trường Name

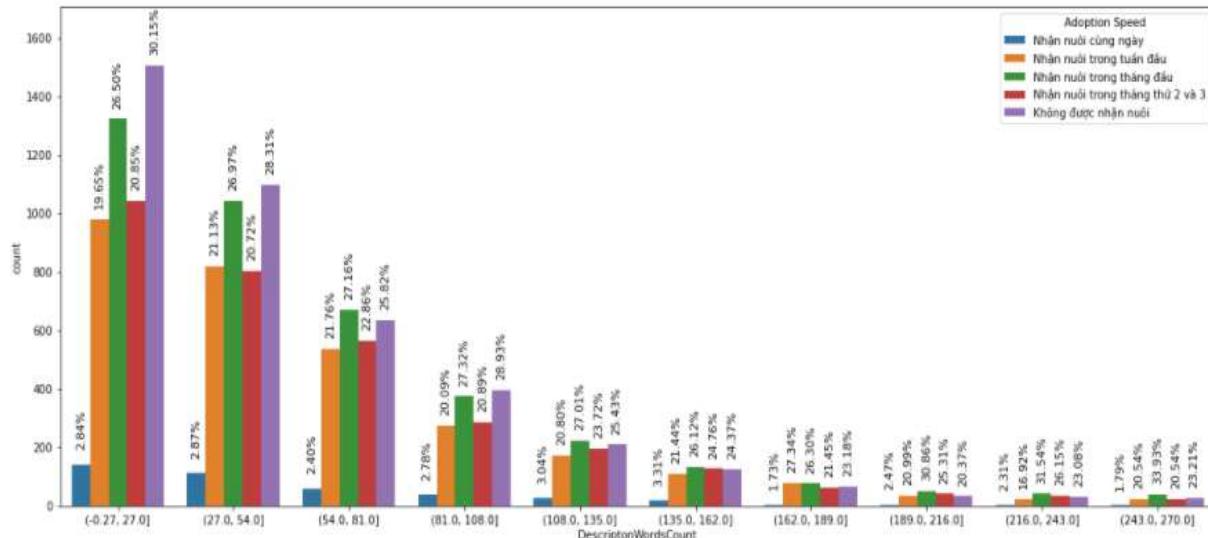
Những tên xuất hiện nhiều nhất như: Kitten, Puppies, Adoption, Girl, Boy, Baby, Black, Brownie.. cho thấy tương tự như phần thông tin mô tả, tên của con vật được cứu hộ được đặt theo các đặc điểm chung nhất hoặc nét ngoại hình nổi bật nhất... Nguyên nhân của điều này là bởi tình trạng của con vật khi được cứu đều rất tệ (bị xua đuổi, bị ngược đãi, bị bỏ đói, suy kiệt và có những vết thương trầm trọng trên cơ thể..) nên những nhân viên cứu hộ trước hết sẽ ưu tiên giúp con vật cải thiện các chỉ số sinh tồn. Mặt khác, lượng động vật cần cứu hộ rất nhiều nên chúng sẽ được theo dõi thông qua các mã định danh (PETID) để tiện cho việc thống kê và

## Dự đoán khả năng được nhận nuôi của động vật từ trạm cứu hộ với dữ liệu đa dạng

việc đặt tên cho con vật được xem là phần quà ý nghĩa dành cho người nhận nuôi con vật đó để đánh dấu thời điểm con vật được nhận nuôi và gia nhập vào gia đình mới.

### **8.2.1. Tương quan giữa độ dài của mô tả với tốc độ con vật được nhận nuôi**

Ta có thể thấy một xu hướng rằng thông tin mô tả càng dài thì cơ hội của con vật được nhận nuôi càng cao.



Hình 8.5. Tương quan giữa Description và Adoptionspeed

Điều này có thể được hiểu rằng: Mọi người muốn biết thêm về con vật trước khi nhận nuôi nó, thật tốt nếu con vật được mô tả với nhiều thông tin để người có ý định nhận nuôi có thể đánh giá sự phù hợp của con vật đó với gia đình mình. Nhóm đánh giá thông số này rất đáng chú ý.

### **8.2.3. Sinh mô hình TF-IDF từ trường Description**

```

vectorizer = TfidfVectorizer(stop_words='english', ngram_range=(4,4), analyzer='char_wb', min_df=10, max_df=0.6, strip_accents='unicode')
X_train = vectorizer.fit_transform(train_desc_df.Description)
X_test = vectorizer.transform(test_desc_df.Description)
y_train = train_desc_df.AdoptionSpeed

save_npz('train_desc_df.npz', X_train)
save_npz('test_desc_df.npz', X_test)

X_train = load_npz('train_desc_df.npz')
X_test = load_npz('test_desc_df.npz')

X_train
<14993x12975 sparse matrix of type '<class 'numpy.float64'>'>
with 2402166 stored elements in Compressed Sparse Row format>

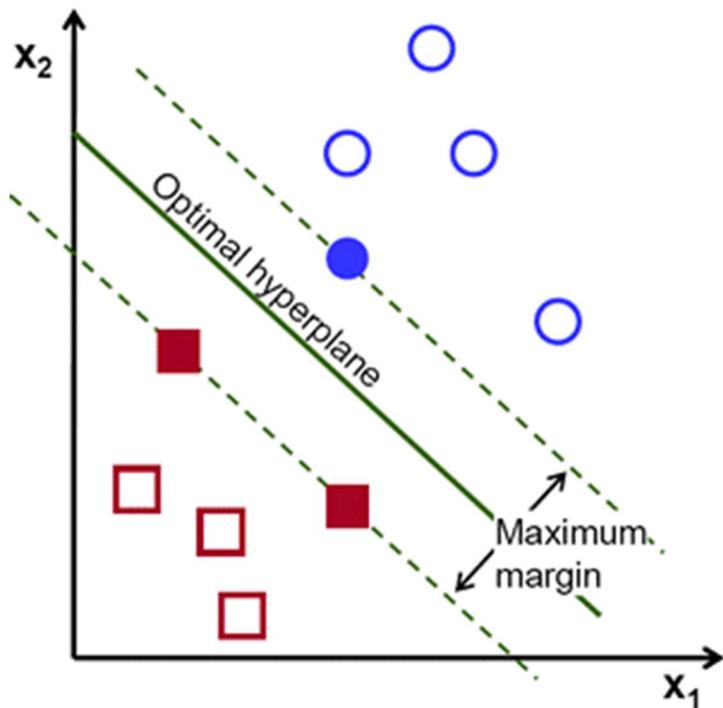
```

Ma trận TF-IDF có 14993 hàng, 12975 cột; mỗi một hàng là mô tả về một con vật; mỗi một cột biểu diễn một cụm từ. Tuy kết quả là dạng ma trận thưa (sparse matrix) nhưng ma trận dữ liệu lớn vẫn khiến tốn dung lượng bộ nhớ và tốn thời gian để load dữ liệu. Do vậy, nhóm lựa chọn load kết quả thu được vào file định dạng .npz để giảm dung lượng.

## 9. XÂY DỰNG VÀ ĐÁNH GIÁ MÔ HÌNH

### 9.1. Các mô hình sử dụng

#### 9.1.1. Support vector machine (SVM)



Hình 9.1 Thuật toán Support Vector Machine

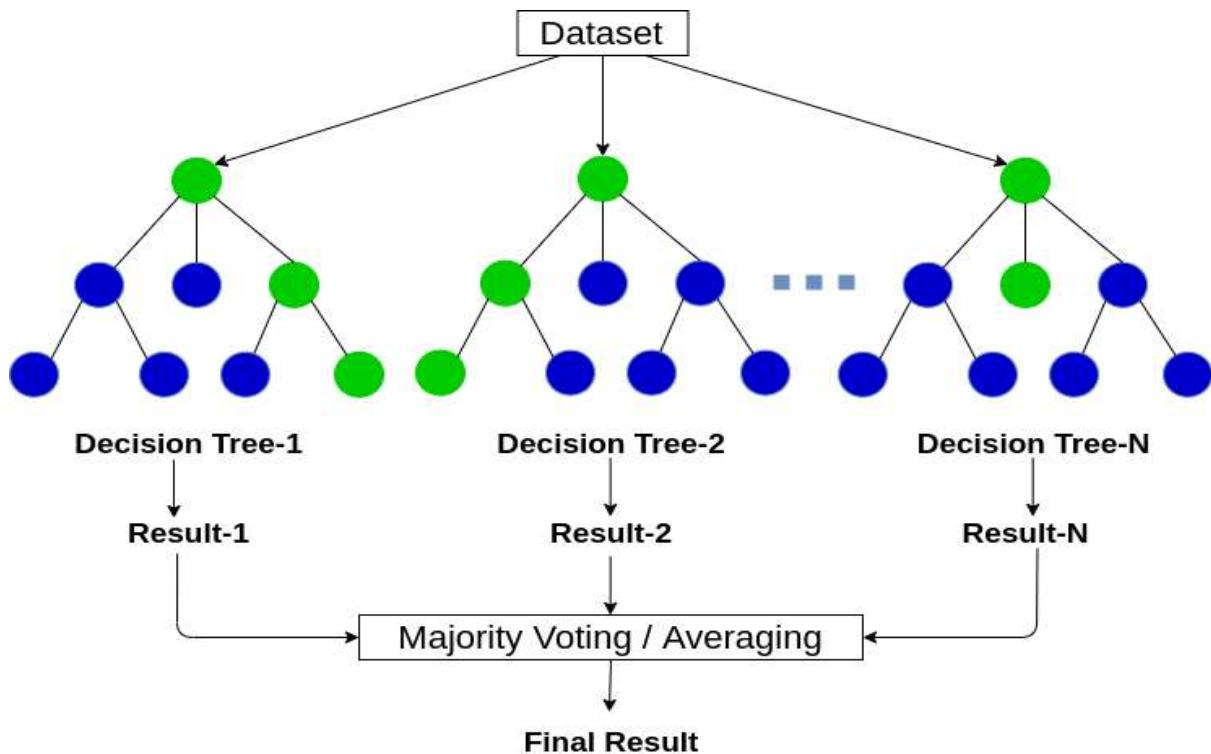
SVM là một thuật toán giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đề quy. Tuy nhiên nó được sử dụng chủ yếu cho việc phân loại. Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (hyperplane) phân chia các lớp. Hyperplane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

Support Vectors hiểu một cách đơn giản là các đối tượng trên đồ thị tọa độ quan sát, Support Vector Machine là một biên giới để chia hai lớp tốt nhất và thỏa mãn các tiêu chí

- Chọn một hyper-plane để phân chia các lớp tốt nhất.
- Xác định khoảng cách lớn nhất từ điểm gần nhất của một lớp nào đó đến đường hyper-plane.
- Cho phép bỏ qua các ngoại lệ và tìm ra hyper-plane có biên giới tối đa.

#### 9.1.2. Random Forest

Phương pháp Random Forest sẽ xây dựng nhiều cây quyết định. Tuy nhiên mỗi cây quyết định sẽ khác nhau có yếu tố ngẫu nhiên. Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định. Thực nghiệm đã chỉ ra rằng phương pháp này hiệu quả hơn việc chỉ dùng một decision tree thông thường với khả năng tổng quát hóa tốt hơn, tránh hiện tượng overfit nhưng đánh đổi bằng việc ta không thể hiểu cơ chế hoạt động của thuật toán này do cấu trúc quá phức tạp của mô hình này — do vậy thuật toán này là một trong những phương thức Black Box — tức ta sẽ bỏ tay vào bên trong và rút ra được kết quả chứ không thể giải thích được cơ chế hoạt động của mô hình. Đó là sự đánh đổi giữa khả năng giải thích và khả năng dự báo.



Hình 9.2. Mô hình thuật toán Random Forest

Yếu tố ngẫu nhiên của Random Forest khi sinh ra mỗi cây trong mô hình thể hiện ở hai điểm

- Mỗi cây quyết định sẽ được học một tập con mẫu khác nhau sinh ngẫu nhiên từ dữ liệu tổng bằng phương pháp bootstraping
- Việc lựa chọn feature tại mỗi nhánh của cây sẽ dựa trên feature cho kết quả tốt nhất trong số các feature được lấy ngẫu nhiên thay vì xét trên toàn bộ tất cả các feature

### 9.1.3. LightGBM

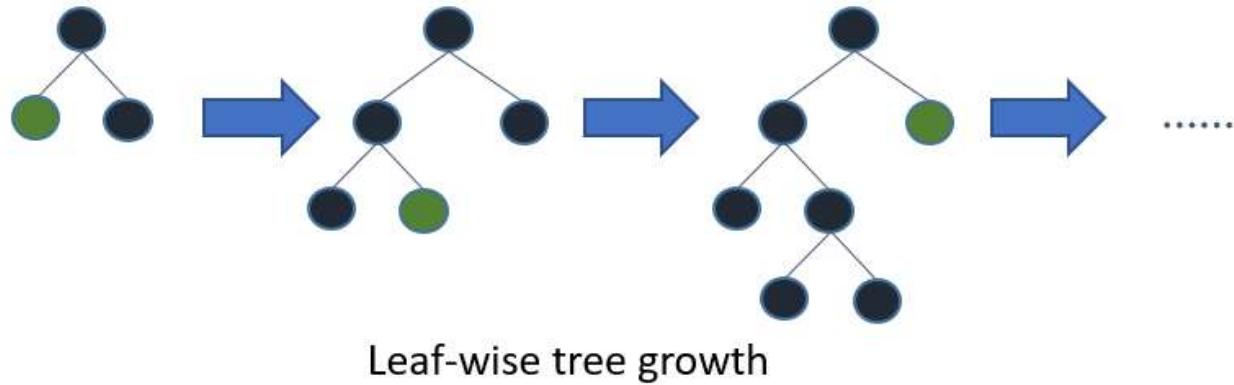
Cùng với XGBoost, LightGBM là một trong 2 framework phổ biến nhất của phương pháp Gradient Boosting. Mặc dù đạt được những kết quả vượt trội, XGBoost gặp một vấn đề là thời gian training khá lâu, đặc biệt với những bộ dữ liệu lớn. Đến tháng 1 năm 2016, Microsoft lần đầu release phiên bản thử nghiệm LightGBM, và LightGBM nhanh chóng thay thế vị trí của XGBoost, trở thành thuật toán ensemble được ưa chuộng nhất.

LightGBM có một số những cải tiến sau đây

- LightGBM sử dụng "histogram-based algorithms" thay thế cho "pre-sort-based algorithms" thường được dùng trong các thuật toán boosting khác để tìm kiếm điểm phân chia trong quá trình xây dựng. Cải tiến này giúp LightGBM tăng tốc độ học, đồng thời làm giảm bộ nhớ cần sử dụng. Thật ra cả xgboost và lightgbm đều sử

dụng histogram-based algorithms, điểm tối ưu của lightgbm so với xgboost là ở 2 thuật toán: GOSS (Gradient Based One Side Sampling) và EFB (Exclusive Feature Bundling) giúp tăng tốc đáng kể trong quá trình tính toán.

- LightGBM phát triển cây dựa trên lá (leaf-wise), trong khi hầu hết các thuật toán Boosting khác (kể cả XGBoost) dựa trên cấp (level-wise). Lá lựa chọn nút để phát triển dựa trên tối ưu toàn bộ cây, trong khi độ sâu tối ưu trên nhánh đang giám sát, do đó, với số nút nhỏ, cây được xây dựng từ lá thường cho kết quả tốt hơn.



Hình 9.3. Thuật toán LightGBM

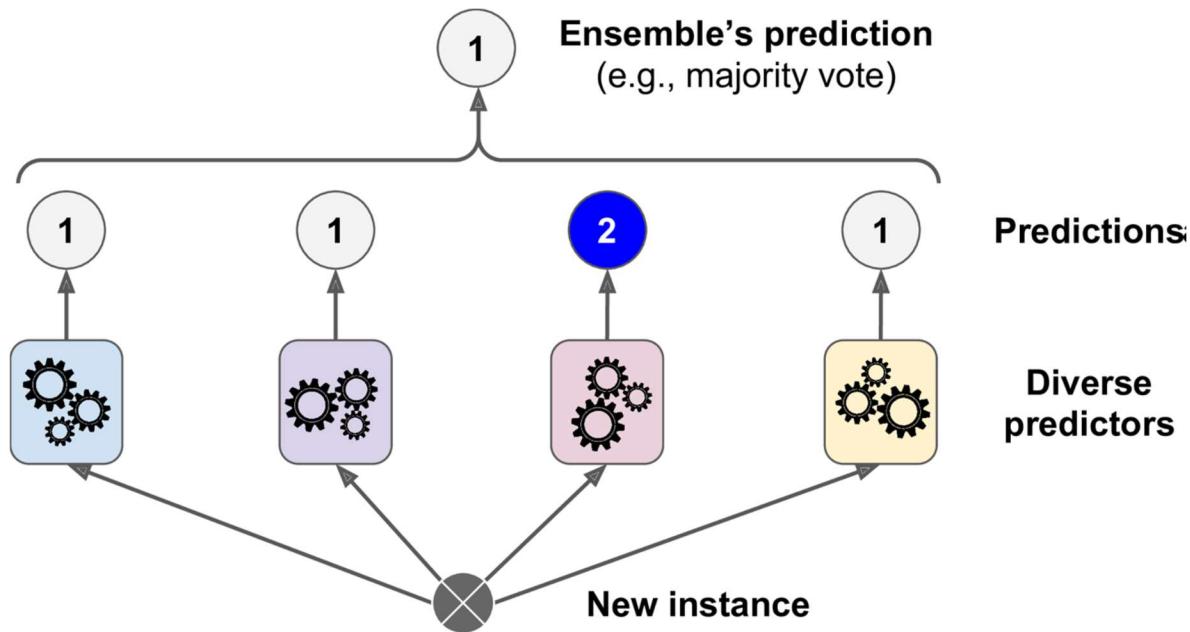
#### 9.1.4. Kết hợp mô hình (ensemble)

Trong machine learning không tồn tại một thuật toán mà luôn tốt cho mọi ứng dụng và mọi tập dữ liệu, vì các thuật toán machine learning thường dựa trên một tập các tham số hoặc một giả thiết nhất định nào đó về phân bố dữ liệu. Vì vậy để tìm được những thuật toán phù hợp cho tập dataset của mình có lẽ cần nhiều thời gian để test các thuật toán khác nhau. Rồi từ đó thực hiện hiệu chỉnh các tham số của thuật toán để thu được độ chính xác cao nhất.

Phương pháp ensemble learning sẽ kết hợp các mô hình với nhau để làm tăng độ chính xác. Ý tưởng của việc kết hợp các mô hình khác nhau xuất phát từ một suy nghĩ hợp lý là: các mô hình khác nhau có khả năng khác nhau, có thể thực hiện tốt nhất các loại công việc khác nhau (subtasks), khi kết hợp các mô hình này với nhau một cách hợp lý thì sẽ tạo thành một mô hình kết hợp mạnh có khả năng cải thiện hiệu suất tổng thể so với việc chỉ dùng các mô hình một cách đơn lẻ.

Trong báo cáo này sẽ sử dụng 3 mô hình Support Vector Machine, Random Forest và LightGBM ở trên để lấy làm mô hình dự báo. Các mô hình này được lấy từ thư viện Sklearn.

Các phương pháp kết hợp hoạt động tốt nhất khi các yếu tố dự đoán càng độc lập với nhau càng tốt. Một cách để có được các bộ phân loại đa dạng là đào tạo chúng bằng các thuật toán rất khác nhau. Điều này làm giảm khả năng mắc cùng một loại lỗi, cải thiện độ chính xác của nhóm.



Hình 9.4. Phương thức ensemble kết hợp các mô hình

Có 2 phương thức kết hợp là hard-voting và soft-voting trong đó

- Đối với hard-voting, lớp dự đoán sẽ dựa trên đa số lớp dự đoán của các mô hình thành viên
- Đối với soft-voting, lớp dự đoán sẽ dựa trên tổng xác suất dự đoán của các mô hình thành viên cho mỗi lớp. Nếu tất cả các bộ phân loại đều có thể ước tính xác suất của lớp (tức là tất cả chúng đều có phương thức `predict_proba()`), thì ta có thể yêu cầu Scikit-Learn dự đoán lớp có xác suất lớp cao nhất, được tính trung bình trên tất cả các bộ phân loại riêng lẻ. Cách này thường đạt được hiệu suất cao hơn so với hard voting vì nó có trọng lượng hơn đối với các phiếu có độ tin cậy cao.

## 9.2. Phương thức đánh giá

### 9.2.1. Phân tách dữ liệu (train-test split)

Trong các bài toán học máy, ta thường chia nhỏ dataset thành các phần nhỏ hơn:

- Training set: Đây thường là một tập dữ liệu có kích thước lớn, được dùng để training mô hình trong quá trình huấn luyện máy học.
- Testing set: là tập dữ liệu dùng để test sau khi máy đã học xong nhằm kiểm chứng xem nó có đạt hiệu quả không
- Validation set: là tập dữ liệu để kiểm thử trong quá trình huấn luyện (khác với Testing set là kiểm thử sau quá trình huấn luyện).

Trong báo cáo, ta sẽ chia tập dữ liệu trong file Training.csv thành 2 phần 80% sẽ sử dụng cho Training, 20% sẽ sử dụng cho Validation. Dữ liệu Testing sẽ lấy từ file Testing.csv



Hình 9.5. Mô tả các tập Training, Validation, Testing

### 9.2.2. Kiểm chứng chéo (Cross validation)

Cross validation là một kỹ thuật lấy mẫu để đánh giá mô hình học máy trong trường hợp dữ liệu không được dòi dào cho lầm.

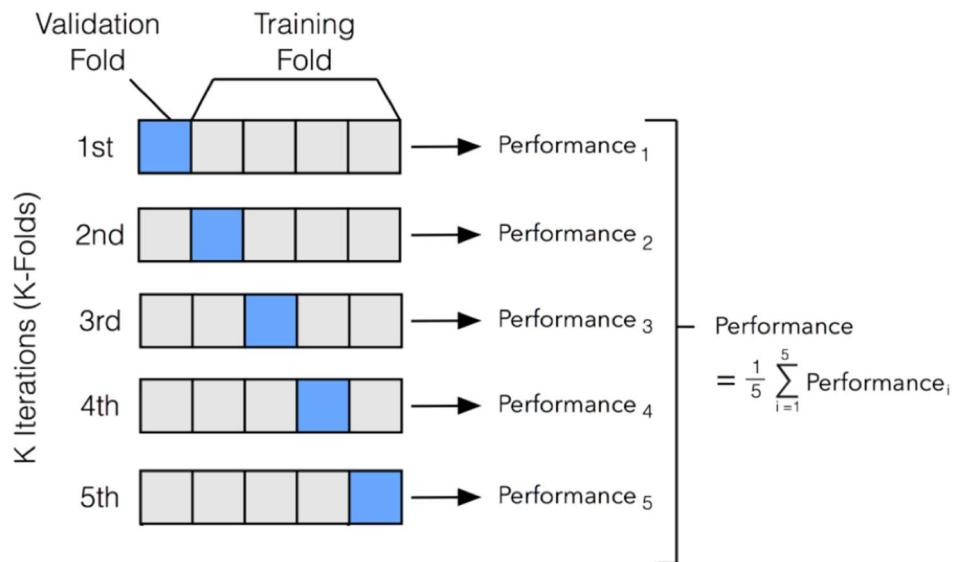
Tham số quan trọng trong kỹ thuật này là k, đại diện cho số nhóm mà dữ liệu sẽ được chia ra. Vì lý do đó, nó được mang tên k-fold cross-validation. Khi giá trị của k được lựa chọn, người ta sử dụng trực tiếp giá trị đó trong tên của phương pháp đánh giá. Ví dụ với k=10, phương pháp sẽ mang tên 10-fold cross-validation.

Kỹ thuật này thường bao gồm các bước như sau:

- Xáo trộn dataset một cách ngẫu nhiên
- Chia dataset thành k nhóm
- Với mỗi nhóm:
  - Sử dụng nhóm hiện tại để đánh giá hiệu quả mô hình
  - Các nhóm còn lại được sử dụng để huấn luyện mô hình
  - Huấn luyện mô hình
  - Đánh giá và sau đó mô hình
- Tổng hợp hiệu quả của mô hình dựa từ các số liệu đánh giá

Một lưu ý quan trọng là mỗi mẫu chỉ được gán cho duy nhất một nhóm và phải ở nguyên trong nhóm đó cho đến hết quá trình. Các tiền xử lý dữ liệu như xây dựng vocabulary chỉ được thực hiện trên tập huấn luyện đã được chia chứ không được thực hiện trên toàn bộ dataset. Việc hủy mô hình sau mỗi lần đánh giá là bắt buộc, tránh trường hợp mô hình ghi nhớ nhãn của tập test trong lần đánh giá trước. Các lỗi thiết lập này dễ xảy ra và đều dẫn đến kết quả đánh giá không chính xác (thường là tích cực hơn so với thực tế).

Kết quả tổng hợp thường là trung bình của các lần đánh giá. Ngoài ra việc bổ sung thông tin về phuơng sai và độ lệch chuẩn vào kết quả tổng hợp cũng được sử dụng trong thực tế.



Hình 9.6. Đánh giá kết quả bằng cross validation

### 9.3. Kết quả đánh giá

Để đánh giá chất lượng mô hình, trong báo cáo này sử dụng 2 phương pháp có sẵn trong thư viện sklearn là tính độ chính xác (accuracy) và chỉ số quadratic kappa score. Kết quả đánh giá cho các mô hình độc lập được trả về như bảng dưới đây.

	Voting	SVM	Random Forest	LightGBM
Accuracy	0.382	0.384	0.379	0.430
Quadratic Kappa	0.300	0.297	0.280	0.379

Bảng 9.1. Kết quả đánh giá các mô hình

Như vậy mô hình LightGBM cho kết quả tốt hơn khá nhiều so với các mô hình còn lại và cả mô hình kết hợp. Chỉ số Quadratic Kappa đạt 0.379 cũng là một con số tương đối tốt. Để dễ hình dung kết quả, nhóm sử dụng thêm chỉ số accuracy truyền thống đạt 43%. Như đã đề cập từ trước do đặc thù bài toán phân loại không chắc chắn tuyệt đối, ví dụ như phân loại ô tô và xe máy, hình ảnh ô tô chắc chắn sẽ phải thuộc lớp ô tô, đối với bài toán này, một con vật có thể thuộc bất cứ lớp nào từ 0 đến 4 nên việc đạt được độ chính xác 43% có thể coi là một kết quả dự đoán khá tốt cho thời gian chờ đợi của vật nuôi

## **10. KẾT LUẬN**

Qua nghiên cứu đề tài cho thấy trước khi bắt đầu giải quyết một bài toán việc chọn đúng thước đo đánh giá kết quả là một việc rất quan trọng, quyết định việc lựa chọn mô hình có phù hợp với nhu cầu thực tế hay không. Đối với bài toán phân loại, ngoài các thước đo truyền thống như accuracy, precision, recall, ROC AUC, ... ta còn có thể sử dụng các thước đo khác như Quadratic Kappa nếu các lớp có mối liên hệ thứ tự.

Ngoài ra hiện nay ngày càng có nhiều thuật toán mới được sử dụng trong việc dự báo, phân loại. Để cải thiện kết quả ta có thể thử nghiệm nhiều thuật toán khác nhau để tìm ra mô hình phù hợp nhất với bộ dữ liệu và bài toán đang giải quyết. Ngoài ra, việc kết hợp các mô hình cũng là một phương pháp đơn giản mà hiệu quả giúp nâng cao chất lượng mô hình.

Bên cạnh việc xây dựng mô hình, phần lớn thời gian giải quyết một bài toán khoa học dữ liệu tập trung vào việc làm sạch dữ liệu và xử lý các biến. Trong khi việc thử nghiệm các mô hình khác nhau mang lại sự chênh lệch không quá nhiều về kết quả, thì việc xử lý dữ liệu giúp cải thiện đáng kể chất lượng dự báo đầu ra với một số công đoạn quan trọng như mã hóa dữ liệu, xử lý dữ liệu thiếu, loại bỏ outlier hay giảm chiều dữ liệu.

Bài toán xác định thời gian con vật chờ nhận nuôi là một dự án có ý nghĩa với cộng đồng. Qua dự án này nhóm mong muốn cung cấp một quy trình hoàn chỉnh cùng phương pháp tiếp cận có hệ thống cho một bài toán Học Máy để giải quyết một vấn đề không chỉ cho bài toán phân loại mà có thể áp dụng cho nhiều bài toán thực tiễn khác trong cuộc sống với đa dạng các kiểu dữ liệu đầu vào.

## TÀI LIỆU THAM KHẢO

1. Bài giảng, tài liệu của giảng viên môn Học máy và khai phá dữ liệu nâng cao.
2. Tài liệu lƣợng dǎn khai phá dữ liệu metadata ảnh được tạo ra bằng Cloud Vision API của Google Cloud.  
<https://cloud.google.com/vision/docs/features-list>
3. Tài liệu lƣợng dǎn xử lý ngôn ngữ tự nhiên bằng Cloud Nature Language của Google Cloud.  
<https://cloud.google.com/vision/docs/features-list>
4. Dan Jurafsky and James H.Martin (3rd ed.draft): Speech and Language Processing.  
<https://web.stanford.edu/~jurafsky/slp3/>