**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN II**

Icon

Description automatically generated with medium confidence**🙞🕮🙜**

**BÁO CÁO CUỐI KỲ**

**HỌC PHẦN: IOT VÀ ỨNG DỤNG**

**ĐỀ TÀI: XÂY DỰNG, ĐÁNH GIÁ MÔ HÌNH MACHINE LEARNING DÙNG PYTHON VÀ SCIKIT-LEARN**

**Giảng viên hướng dẫn: Th.S Đàm Minh Lịnh**

**Nhóm thực hiện: Nhóm 33**

1. **Cao Thanh Nhàn N19DCCN125**
2. **Lê Thị Thùy Trang N19DCCN207**

**TP. HCM, 12/2022**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN II**

Icon

Description automatically generated with medium confidence**🙞🕮🙜**

**BÁO CÁO CUỐI KỲ**

**HỌC PHẦN: IOT VÀ ỨNG DỤNG**

**ĐỀ TÀI: XÂY DỰNG, ĐÁNH GIÁ MÔ HÌNH MACHINE LEARNING DÙNG PYTHON VÀ SCIKIT-LEARN**

**Giảng viên hướng dẫn: Th.S Đàm Minh Lịnh**

**Nhóm thực hiện: Nhóm 33**

1. **Cao Thanh Nhàn N19DCCN125**
2. **Lê Thị Thùy Trang N19DCCN207**

**TP. HCM, 12/2022**

# LỜI CẢM ƠN

Đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến giảng viên bộ môn – Thầy Đàm Minh Lịnh đã dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học “IOT và ứng dụng” của thầy, chúng em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này.

Bộ môn “IOT và ứng dụng” là môn học thú vị, vô cùng bổ ích và có tính thực tế cao, đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, với sự thiếu sót về kiến thức và kinh nghiệm cũng như thời gian có hạn nên khó tránh những hạn chế, chúng em mong nhận được sự đóng góp ý kiến của thầy để chúng em có thể học hỏi thêm được những kiến thức và kinh nghiệm giúp ích cho quá trình học tập và công việc sau này.

Chúng em xin chân thành cảm ơn!

*Thành phố Hồ Chí Minh, tháng 12 năm 2022*

**Nhóm sinh viên thực hiện**

# **MỤC LỤC**

[LỜI CẢM ƠN i](#_Toc123058316)

[MỤC LỤC ii](#_Toc123058317)

[PHẦN 1. CƠ SỞ LÝ THUYẾT 1](#_Toc123058318)

[1.1. Machine Learning: 1](#_Toc123058319)

[1.2. Thư viện Scikit-learn: 1](#_Toc123058320)

[1.3. Perceptron nhiều lớp (Multilayer Perceptron Classifier – MLP): 2](#_Toc123058321)

[PHẦN 2. THỰC NGHIỆM: 4](#_Toc123058322)

[2.1. Tiền xử lý dữ liệu: 4](#_Toc123058323)

[2.1.1. Ý nghĩa: 4](#_Toc123058324)

[2.1.2. Phương pháp thực hiện: 4](#_Toc123058325)

[2.2. Train model: 6](#_Toc123058326)

[2.2.1. Import các thư viện sau: 6](#_Toc123058327)

[2.2.2. Khai báo và load dữ liệu từ file Feature.pickle 7](#_Toc123058328)

[2.2.3. Train model 7](#_Toc123058329)

[2.2.4. In ra màn hình tỉ lệ dữ liệu mẫu và dữ liệu test bao nhiêu phần trăm: 9](#_Toc123058330)

[2.2.5. Lưu lại model 10](#_Toc123058331)

[2.2.6. Mã hóa nhãn theo nhiều cột: 10](#_Toc123058332)

[2.2.7. Bảng báo cáo độ chính xác dữ liệu dùng để test của Dog và Cat: 10](#_Toc123058333)

[2.2.8. Đánh giá: 11](#_Toc123058334)

[2.3. Demo: 12](#_Toc123058335)

[2.3.1. Import các thư viện: 12](#_Toc123058336)

[2.3.2. Khai báo labels 13](#_Toc123058337)

[2.3.3. Xây dựng giao diện 13](#_Toc123058338)

[2.3.4. Kết quả: 14](#_Toc123058339)

# PHẦN 1. CƠ SỞ LÝ THUYẾT

## Machine Learning:

Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1 - động cơ hơi nước, 2 - năng lượng điện, 3 - công nghệ thông tin). Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind,…chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning.

***Machine Learning (ML)*** hay máy học là một nhánh của Trí Tuệ Nhân Tạo (AI), nó là một lĩnh vực nghiên cứu cho phép máy tính có khả năng cải thiện chính bản thân chúng dựa trên dữ liệu mẫu (training data) hoặc dựa vào kinh nghiệm (những gì đã được học). Machine Learning có thể tự dự đoán hoặc đưa ra quyết định mà không cần được lập trình cụ thể.

Bài toán Machine Learning thường được chia làm hai loại là dự đoán (Prediction) và phân loại (Classification). Các bài toán dự đoán như dự đoán giá nhà, giá xe… Các bài toán phân loại như nhận diện chữ viết tay, nhận diện đồ vật…

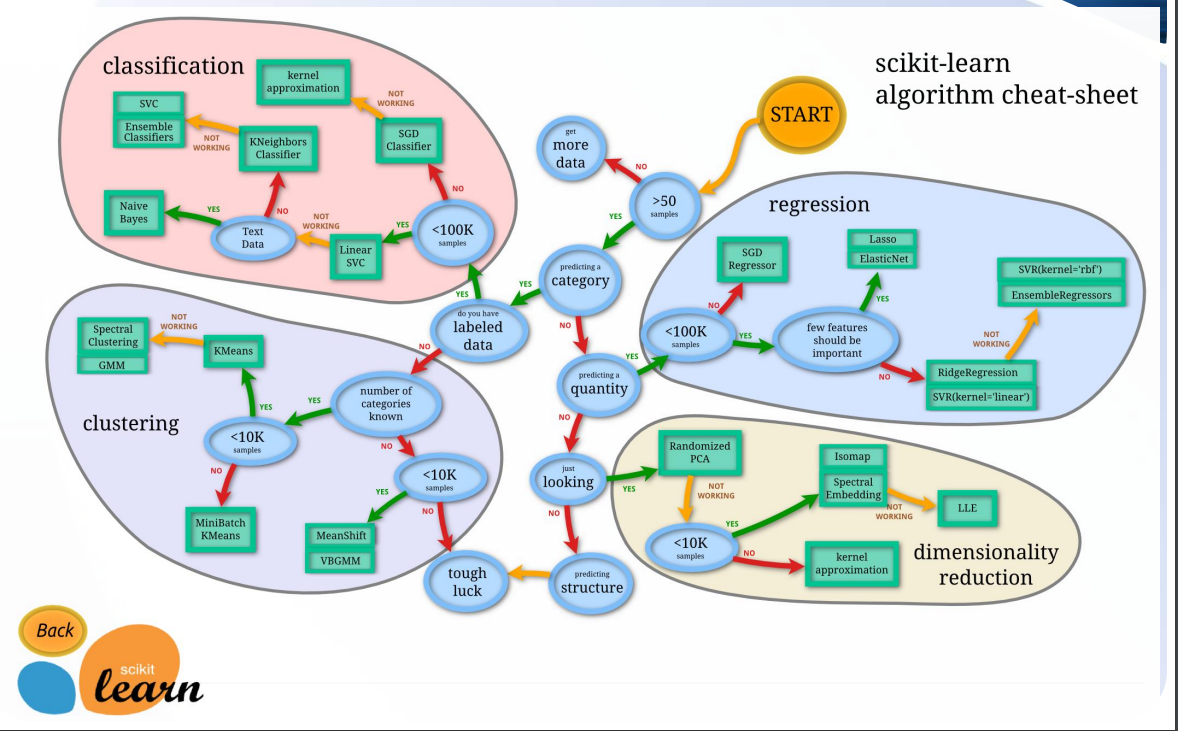
Xét theo phương thức học, các thuật toán ML được chia làm bốn nhóm, bao gồm “Học có giám sát” (Supervised Learning), “Học không giám sát” (Unsupervised Learning), “Học bán giám sát” (hay học kết hợp - Semi-supervised Learning) và “Học tăng cường” (Reinforcement Learning).

## Thư viện Scikit-learn:

***Scikit-Learn (Sklearn)*** được phát triển bởi David Cournapeau vào năm 2007 như một dự án Google Summer of Code. Tên của dự án bắt nguồn từ khái niệm rằng nó là một "SciKit" (Bộ công cụ SciPy), một phần mở rộng của bên thứ ba được phát triển và phân phối riêng cho SciPy . Cơ sở mã ban đầu sau đó được viết lại bởi các nhà phát triển khác. Năm 2010, những người đóng góp Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort và Vincent Michel, từ Viện Nghiên cứu Khoa học Máy tính và Tự động hóa Pháp ở Saclay, Pháp, đã lãnh đạo dự án và phát hành phiên bản công khai đầu tiên của thư viện vào ngày 1 tháng 2 năm 2010 - Thư viện scikits. Vào năm 2019, người ta đã lưu ý rằng scikit-learning là một trong những thư viện máy học phổ biến nhất trên GitHub.

Scikit-learn (Sklearn) là thư viện mã nguồn mở mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện này hỗ trợ các thuật toán hiện đại như KNN, XGBoost, random forest, SVM và một số thuật toán khác. Scikit-learn cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.

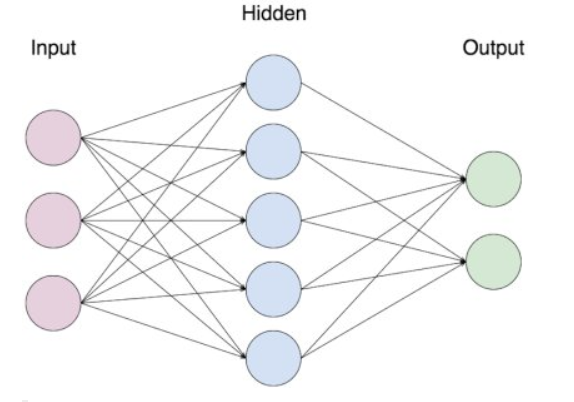
Sau đâu là một số nhóm thuật toán được xây dựng bởi thư viện Scikit-learn:

* Clustering: Nhóm thuật toán Phân cụm dữ liệu không gán nhãn. Ví dụ thuật toán KMeans
* Cross Validation: Kiểm thử chéo, đánh giá độ hiệu quả của thuật toán học giám sát sử dụng dữ liệu kiểm thử (validation data) trong quá trình huấn luyện mô hình.
* Datasets: Gồm nhóm các Bộ dữ liệu được tích hợp sẵn trong thư viện. Hầu như các bộ dữ liệu đều đã được chuẩn hóa và mang lại hiêu suất cao trong quá trình huấn luyện như iris, digit, ...
* Dimensionality Reduction: Mục đích của thuật toán này là để Giảm số lượng thuộc tính quan trọng của dữ liệu bằng các phương pháp như tổng hợp, biểu diễn dữ liệu và lựa chọn đặc trưng. Ví dụ thuật toán PCA (Principal component analysis).
* Ensemble methods: Các Phương pháp tập hợp sử dụng nhiều thuật toán học tập để có được hiệu suất dự đoán tốt hơn so với bất kỳ thuật toán học cấu thành nào.
* Feature extraction: Trích xuất đặc trưng. Mục đích là để định nghĩa các thuộc tình với dữ liệu hình ảnh và dữ liệu ngôn ngữ.
* Feature selection: Trích chọn đặc trưng. Lựa chọn các đặc trưng có ý nghĩa trong việc huấn luyện mô hình học giám sát.
* Parameter Tuning: Tinh chỉnh tham số. Các thuật toán phục vụ việc lựa chọn tham số phù hợp để tối ưu hóa mô hình.
* Manifold Learning: Các thuật toán học tổng hợp và Phân tích dữ liệu đa chiều phức tạp.
* Supervised Models: Học giám sát. Mảng lớn các thuật toán học máy hiện nay. Ví dụ như linear models, discriminate analysis, naive bayes, lazy methods, neural networks, support vector machines và decision trees.

Hình 1.1. Các thuật toán học máy của thư viện Scikit-learn

## Perceptron nhiều lớp (Multilayer Perceptron Classifier – MLP):

***Multilayer Perceptron (Perceptron nhiều lớp)*** là một thuật toán học máy có giám sát (Machine Learning) thuộc lớp Mạng nơ-ron nhân tạo, là tập hợp của các perceptron chia làm nhiều nhóm, mỗi nhóm tương ứng với một layer. Trong hình dưới ta có một ANN với 3 lớp: Input layer (lớp đầu vào), Output layer (lớp đầu ra) và Hidden layer (lớp ẩn). Thông thường khi giải quyết một bài toán ta chỉ quan tâm đến input và output của một model, do vậy trong MLP nói riêng và ANN nói chung ngoài lớp Input và Output ra thì các lớp neuron ở giữa được gọi chung là Hidden. Thuật toán này về cơ bản được đào tạo trên dữ liệu để học một hàm phi tuyến tính nhằm mục đích phân loại hay hồi quy, với một tập hợp các tính năng và một biến mục tiêu (ví dụ: nhãn). Bài báo cáo này tập trung vào trường hợp phân loại.



Hình 1.2. Multi-layer Perception

# PHẦN 2. THỰC NGHIỆM:

## Tiền xử lý dữ liệu:

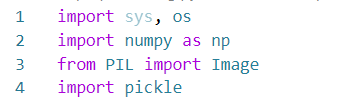
### Ý nghĩa:

Tiền xử lý dữ liệu là một bước rất quan trọng trong việc giải quyết bất kỳ vấn đề nào trong lĩnh vực Học Máy. Hầu hết các bộ dữ liệu được sử dụng trong các vấn đề liên quan đến Học Máy cần được xử lý, làm sạch và biến đổi trước khi một thuật toán Học Máy có thể được huấn luyện trên những bộ dữ liệu này

Ở bài báo cáo này, chúng em sử dụng bộ dữ liệu hình ảnh AnimalFace tại Kaggel.

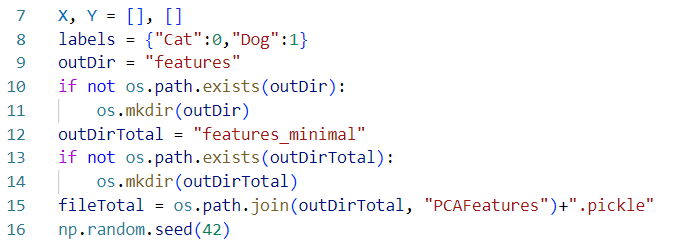
### Phương pháp thực hiện:

1. ***Import các thư viện:***



* Module os: Cho phép thao tác với tệp và thư mục.
* Module np: **Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần.**
* **Module Image: Cho phép xử lý hình ảnh.**
* **Module pickel: Được sử dụng để thực hiện chuyển đổi các cấu trúc đối tượng Python sang một dạng byte để có thể lưu trữ trên ổ đĩa hoặc được gửi qua mạng. Sau đó, luồng ký tự có thể được truy xuất và chuyển đổi trở lại sáng dạng đối tượng ban đầu trong Python.**

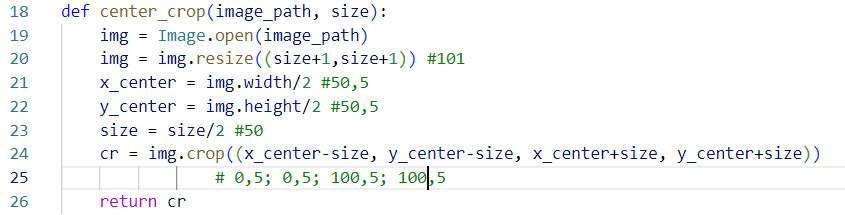
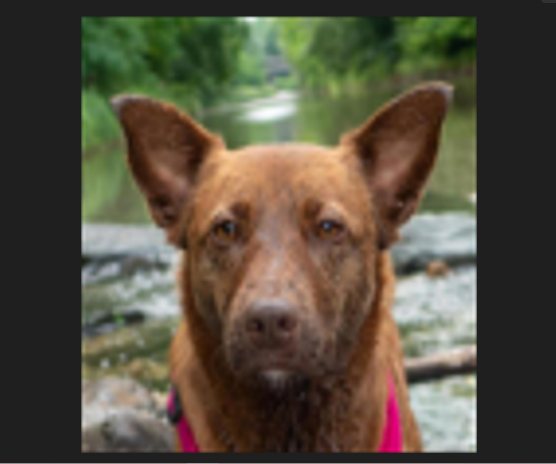
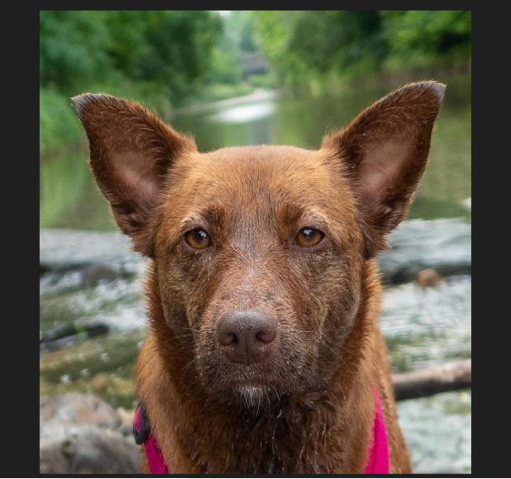
1. ***Khai báo biến và tạo thư mục :***



* Mảng một chiều: Để lưu các giá trị tải lên từ file pickle.
* labels: Lưu tên nhãn kèm theo giá trị tương ứng.
* outDir: Thư mục chứa file đặc trưng pickle của Cat và Dog.
* outDirTotal: Thư mục chứa file tổng pickle của Cat và Dog.
* fileTotal: File tổng pickle của Cat và Dog.
* np.random.seed(42): một hàm được coi như giúp khởi tạo các bộ sinh số ngẫu nhiên (random generator). Biến số trong seed thường là một số nguyên không âm 32 bit.

1. ***Xử lý ảnh trong dataset Cat và Dog***:

Load ảnh mẫu trong thư mục “AnimalFace”, tiếp đến sử dụng hàm kết hợp resize() và crop() để thay đổi kích thước ảnh theo tỉ lệ 100x100 nhằm làm mịn ảnh.



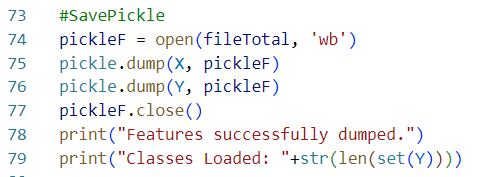
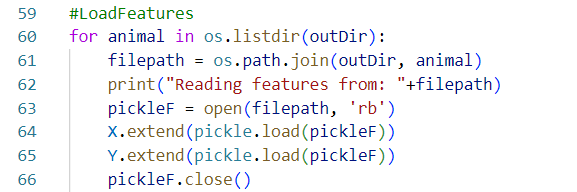
*Hình 2.1 Trước và sau khi thay đổi kích thước ảnh*

Chuyển ảnh về màu xám theo hệ màu RGB, tiến hành trích xuất màu trên ảnh có giá trị từ 0 đến 1, tiếp theo thêm vào mảng kèm theo nhãn của ảnh là Dog hay Cat. Sau khi thực hiện lấy dữ liệu của ảnh xong, dữ liệu sẽ được lưu vào file pickle theo từng file riêng Cat và Dog.



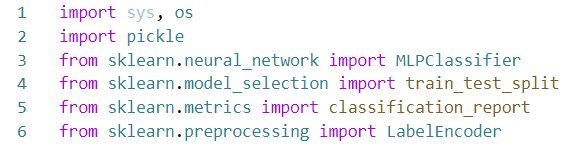
1. ***Lưu dữ liệu vào file chuẩn bị cho quá trình train***

Load dữ liệu từ các file riêng Cat và Dog sau đó gộp lại và lưu vào file chung Features.pickle để tiến hành chia dữ liệu nhằm train và test.



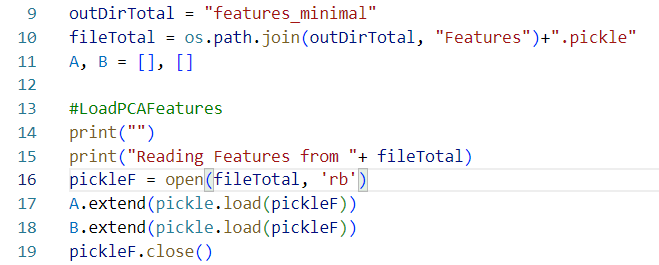
## Train model:

### Import các thư viện sau:



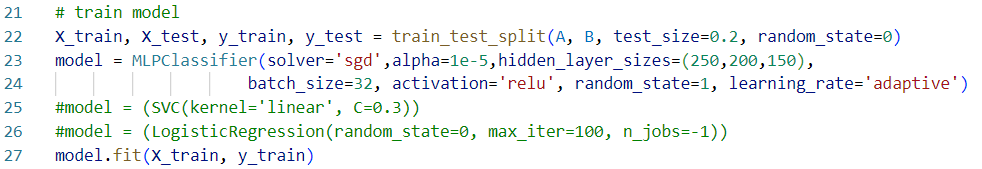
* Module os: Cho phép thao tác với tệp và thư mục.
* Module pickel: Được sử dụng để thực hiện chuyển đổi các cấu trúc đối tượng Python sang một dạng byte để có thể lưu trữ trên ổ đĩa hoặc được gửi qua mạng. Sau đó, luồng ký tự có thể được truy xuất và chuyển đổi trở lại sáng dạng đối tượng ban đầu trong Python.
* Model MLPClassifier: Dùng huấn luyện dữ liệu.
* train\_test\_split: Dùng để chia 80% huấn luyện dữ liệu, 20% còn lại dùng để test
* classification\_report: Bảng báo cáo độ chính xác mẫu test sau khi huấn luyện xong.
* LabelEncoder: Mã hóa nhãn trên nhiều cột.

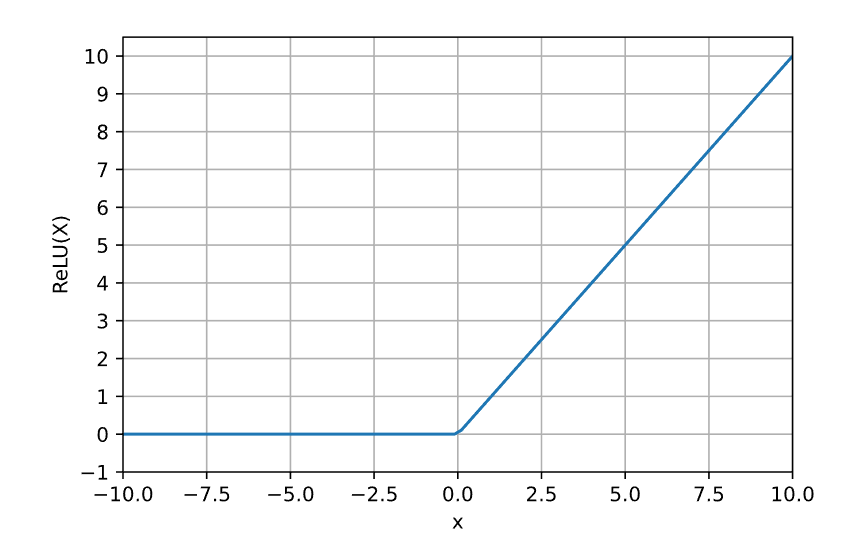
### ****Khai báo và load dữ liệu từ file Feature.pickle****

******Khai báo 2 mảng A và B để lưu lại dữ liệu được load ra từ file Feature.pickle. Mảng A lưu lại đặc trưng, mảng B lưu lại nhãn tương ứng.

### ****Train model****

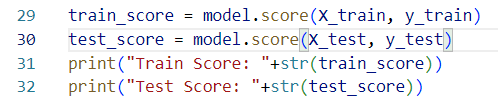
Bắt đầu chia 80% dữ liệu mẫu dùng để huấn luyện và 20% còn lại dùng để test. Sử dụng MLPClassifier thuộc thư viện “sklearn.neural\_network.” để train với các thông số sau:

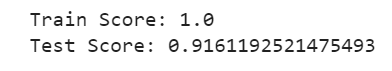
**

* solver = ‘sgd’: bộ giải để tối ưu hóa trọng lượng đề cập đến việc giảm độ dốc ngẫu nhiên.
* alpha = 1e-5: Alpha là một tham số cho thuật ngữ chính quy hóa, hay còn gọi là thuật ngữ hình phạt, chống lại việc trang bị quá mức bằng cách hạn chế kích thước của trọng số. Việc tăng alpha có thể khắc phục phương sai cao (một dấu hiệu của việc trang bị quá mức) bằng cách khuyến khích các trọng số nhỏ hơn, dẫn đến biểu đồ ranh giới quyết định xuất hiện với độ cong nhỏ hơn. Tương tự như vậy, việc giảm alpha có thể khắc phục độ lệch cao (dấu hiệu của sự thiếu phù hợp) bằng cách khuyến khích các trọng số lớn hơn, có khả năng dẫn đến ranh giới quyết định phức tạp hơn.
* hidden\_layer\_sizes = (250,200,150): Một lớp ẩn trong một mạng lưới thần kinh nhân tạo là một lớp ở giữa các lớp đầu vào và đầu ra lớp, nơi tế bào thần kinh nhân tạo mất trong một bộ đầu vào trọng và tạo ra một đầu ra thông qua một chức năng kích hoạt. Nó là một phần tiêu biểu của gần như bất kỳ mạng thần kinh, trong đó kỹ sư mô phỏng các loại hoạt động mà đi vào trong não của con người.Mô hình có 3 lớp ẩn, lớp ẩn đầu có 250 đơn vị, lớp ẩn hai có 200 đơn vị và lớp ẩn cuối cùng có 150 đơn vị.
* batch\_size = 32: Số lượng mẫu dữ liệu trong một lần huấn luyện.
* activation = ‘relu’: Hàm kích hoạt ‘relu’ đơn giản lọc các giá trị < 0.
* random\_state = 1: là siêu tham số (hyperparameter) được sử dụng để thiết lập seed cho trình tạo ngẫu nhiên (random generator). Do bản chất của việc tách (split) dữ liệu thành tập dữ liêu huấn luyện (traning data) và tập dữ liệu thử nghiệm (testing set) là ngẫu nhiên, bạn sẽ nhận được các dữ liệu khác nhau được gán cho tập dữ liệu huấn luyện và tập dữ liệu thử nghiệm. 1 là mã giả ngẫu nhiên.
* learning\_state: Giữ tốc độ học tập không đổi thành 'learning\_rate\_init' miễn là tổn thất đào tạo tiếp tục giảm. Mỗi khi hai kỷ nguyên liên tiếp không giảm được tổng tổn thất đào tạo hoặc không tăng được điểm xác nhận ít nhất nếu 'early\_stopping' được bật, tốc độ học tập hiện tại được chia cho 5. Chỉ được sử dụng khi solver='sgd'.

Sau đó, sử dụng phương thức .fit() để train model trên tập dataset và label đã cho.

### ****In ra màn hình tỉ lệ dữ liệu mẫu và dữ liệu test bao nhiêu phần trăm**:**

******Sử dụng phương thức .score() để xem độ chính xác của train và test.

Tỉ lệ nhận biết và phân loại Dog và Cat: 100% đối với dữ liệu đã được huấn luyện và gần 92% đối với dữ liệu dùng để test, độ chính xác khá cao.

### Lưu lại model

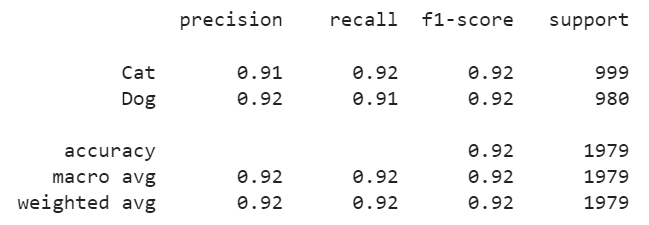
Lưu lại model vào file finalized\_model.sav để có thể dùng trong file test

******

### Mã hóa nhãn theo nhiều cột:

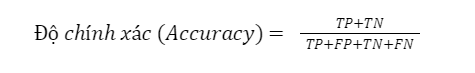
### Bảng báo cáo độ chính xác dữ liệu dùng để test của Dog và Cat:

Tỉ lệ nhận biết ảnh Cat được 91% và 92% đối với ảnh Dog.



### Đánh giá:

Dựa trên bộ dữ liệu thu thập được, hiệu suất trong phân loại dữ liệu được đánh giá bởi độ chính xác của trình phân loại cho tất cả các lớp theo các công thức:



*Trong đó:*

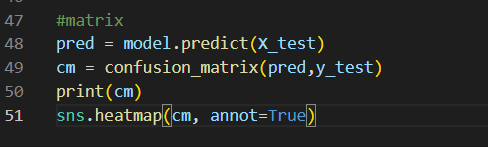
*True Positive (TP): số lượng điểm của lớp positive được phân loại đúng là positive.*

*True Negative (TN): số lượng điểm của lớp negative được phân loại đúng là negative.*

*False Positive (FP): số lượng điểm của lớp negative bị phân loại nhầm thành positive.*

*False Negative (FN): số lượng điểm của lớp positiv bị phân loại nhầm thành negative.*

Ta sử dụng phương thức confusion\_matrix() để xác định các giá trị trên



Từ đó có matrix sau:



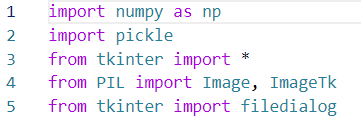
Trong đó:

|  |  |
| --- | --- |
| TP | 912 |
| FP | 91 |
| FN | 87 |
| TN | 889 |

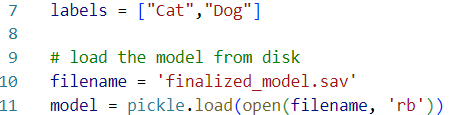
Theo công thức trên ta có kết quả đánh giá độ chính xác của mô hình đạt xấp xỉ **91%.**

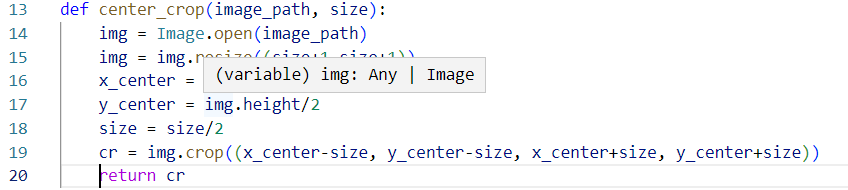
## Demo:

### Import các thư viện:

* Module np: Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần.
* Module pickel: Được sử dụng để thực hiện chuyển đổi các cấu trúc đối tượng Python sang một dạng byte để có thể lưu trữ trên ổ đĩa hoặc được gửi qua mạng. Sau đó, luồng ký tự có thể được truy xuất và chuyển đổi trở lại sáng dạng đối tượng ban đầu trong Python.
* Thư viện tkinter: Là giao diện Python cho bộ công cụ Tk GUI.
* Module Image: Cho phép xử lý hình ảnh.

### ****Khai báo labels****

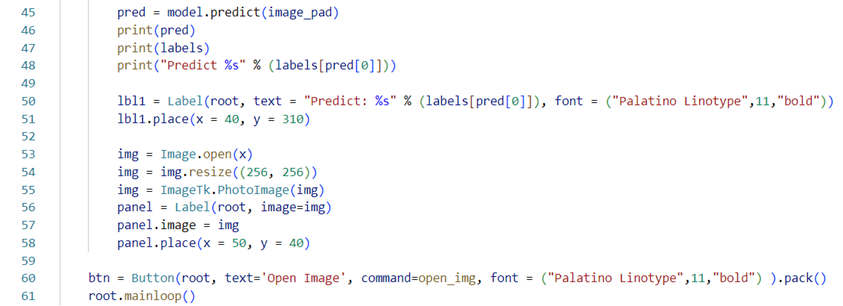
****Khai báo 2 labels Cat và Dog, đưa ra kết quả dự đoán ảnh test, load file model ta đã huấn luyện tiến hành dự đoán ảnh test có khớp với model hay không**

**Tương tự với tiền xử lý dữ liệu, ta sẽ khởi tạo hàm nhằm xử lý ảnh test.**

### Xây dựng giao diện

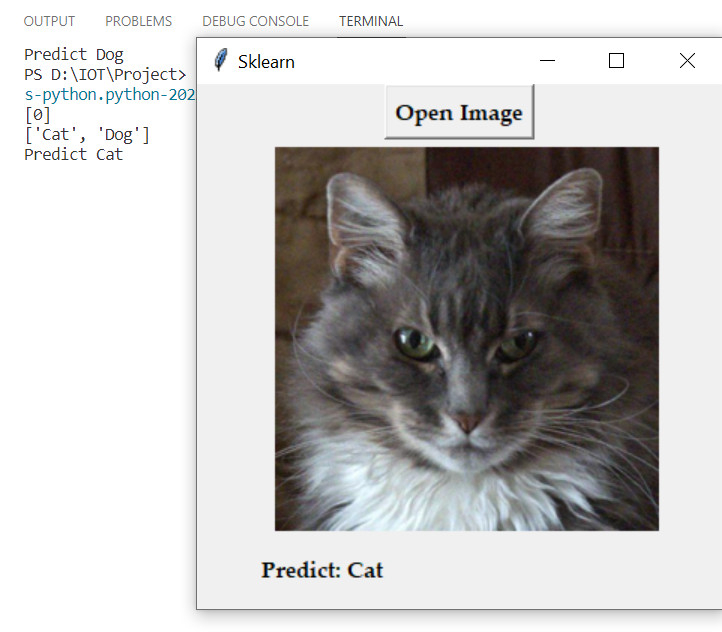
Chọn ảnh mình muốn kiểm tra từ ổ đĩa, tiến hành trình tự xử lý ảnh như giai đoạn tiền xử lý dữ liệu, chuyển ảnh về màu xám theo gam màu RGB, tiến hành trích xuất màu trên ảnh đưa về giá trị từ 0 đến 1

Lấy dữ liệu mới vừa có được từ ảnh test, ta sẽ đưa dữ liệu vào model dự đoán đó là Dog hay Cat, show ảnh mình cần dự đoán đồng thời kèm theo nhãn đã đoán được là Dog hay Cat trên màn hình.



### Kết quả:

Sau khi click “Open Image”, chọn 1 ảnh từ ổ đĩa và kết quả sẽ trả lại Nhãn của ảnh. Ví dụ như khi input 1 ảnh Cat, sẽ hiển thị trên console và giao diện như sau:



Tương tự như với ảnh của Dog

