**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



**BÁO CÁO PROJECT II**

**Đề tài:**

**Dự đoán giá vàng và nhận diện cảm xúc khuôn mặt**

**Sinh viên thực hiện:** Ngô Đức Quang Anh – 20215259

**Giáo viên hướng dẫn:** PGS.TS Trịnh Văn Loan

**Mã lớp học:** 738753 - IT3930

**Hà Nội, 7/2024**

Mục lục

[**Tổng quan về đề tài** 3](#_Toc171374263)

[**Chương 1.Lập trình Python** 5](#_Toc171374264)

[**Chương 2.Tìm hiểu về thuật toán học máy** 9](#_Toc171374265)

[**1.Thuật toán Random Forest** 9](#_Toc171374266)

[**2.Transfer Learning** 10](#_Toc171374267)

[**Chương 3.Bài toán dự đoán giá vàng** 12](#_Toc171374268)

[**Bước 1: Thu Thập Dữ Liệu** 12](#_Toc171374269)

[**Bước 2: Khám Phá và Xử Lý Dữ Liệu** 12](#_Toc171374270)

[**Bước 3: Phân Tích Dữ Liệu** 14](#_Toc171374271)

[**Bước 4: Chuẩn bị dữ liệu** 15](#_Toc171374272)

[**Bước 5: Xây dựng mô hình** 17](#_Toc171374273)

[**Bước 6: Đánh giá mô hình** 17](#_Toc171374274)

[**Bước 7: Trực quan hóa kết quả** 18](#_Toc171374275)

[**Bước 8: Triển khai** 18](#_Toc171374276)

[**Chương 4.Bài toán nhận diện cảm xúc** 20](#_Toc171374277)

[**Bước 1: Thu Thập Dữ Liệu và đọc ảnh** 20](#_Toc171374278)

[**Bước 2.Tạo dữ liệu huấn luyên** 23](#_Toc171374279)

[**Bước 3: Chuẩn bị mô hình** 24](#_Toc171374280)

[**Bước 4: Xây dựng mô hình mới** 25](#_Toc171374281)

[**Bước 5: Huấn luyện mô hình** 26](#_Toc171374282)

[**Bước 6: Nhận diện cảm xúc từ ảnh mới** 27](#_Toc171374283)

[**Chương 5.Kết luận** 32](#_Toc171374284)

[**Tài liệu tham khảo** 33](#_Toc171374285)

# **Tổng quan về đề tài**

Dự án này gồm 2 dự án con : Dự đoán giá vàng và nhận diện cảm xúc khuôn mặt dựa trên hình ảnh.

**1. Dự Đoán Giá Vàng**

**Mô Tả Bài Toán**

Dự đoán giá vàng là một trong những bài toán quan trọng trong lĩnh vực tài chính và đầu tư. Giá vàng biến động hàng ngày dựa trên nhiều yếu tố như tình hình kinh tế toàn cầu, chính sách tiền tệ, cung và cầu, cũng như các sự kiện chính trị. Việc dự đoán chính xác giá vàng giúp các nhà đầu tư đưa ra quyết định đúng đắn, từ đó tối ưu hóa lợi nhuận và giảm thiểu rủi ro.

**Phương Pháp Sử Dụng**

* **Thuật Toán AI: Random Forest**
  + **Random Forest**: Đây là một thuật toán học máy thuộc nhóm ensemble learning, sử dụng nhiều cây quyết định để đưa ra dự đoán. Random Forest có khả năng xử lý dữ liệu không đồng nhất và có độ phức tạp cao, giúp cải thiện độ chính xác của mô hình dự đoán.
  + **Decision Trees**: Là thành phần cơ bản của Random Forest, Decision Trees đơn giản nhưng mạnh mẽ, cho phép mô hình hóa các quyết định dựa trên các thuộc tính đầu vào và các quy tắc logic.

**2. Nhận Diện Cảm Xúc Khuôn Mặt**

**Mô Tả Bài Toán**

Nhận diện cảm xúc khuôn mặt là một bài toán thuộc lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Bài toán này nhằm phân loại các biểu cảm khuôn mặt của con người thành các trạng thái cảm xúc khác nhau như vui, buồn, giận dữ, ngạc nhiên, sợ hãi, và bình tĩnh. Nhận diện cảm xúc khuôn mặt có nhiều ứng dụng trong các lĩnh vực như an ninh, chăm sóc sức khỏe tâm lý, và giao diện người-máy.

**Phương Pháp Sử Dụng**

* **Transfer Learning**: Đây là một kỹ thuật trong học sâu, cho phép sử dụng một mô hình đã được huấn luyện trên một tập dữ liệu lớn và chuyển giao kiến thức cho một bài toán cụ thể. Transfer Learning giúp tiết kiệm thời gian và tài nguyên huấn luyện, đồng thời cải thiện hiệu quả mô hình.
  + **Pre-trained Models**: Các mô hình đã được huấn luyện như MobileNetV2 sẽ được sử dụng và tinh chỉnh trên dữ liệu cảm xúc khuôn mặt.

# **Chương 1.Lập trình Python**

1.TỔNG QUAN

- Python là một ngôn ngữ lập trình bậc cao, được ra mắt vào năm 1991. Python nổi bật với cú pháp rõ ràng, dễ học và dễ hiểu hơn nhiều so với các ngôn ngữ lập trình khác như C++, Java hay PHP. Nó giúp các lập trình viên mới tiếp cận nhanh chóng và các chuyên gia phát triển phần mềm hiệu quả. Ngôn ngữ này hỗ trợ nhiều phong cách lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm và lập trình thủ tục.

-Một ưu điểm quan trọng của Python là tính tương thích và tích hợp tốt với các ngôn ngữ lập trình và công nghệ khác. Python cũng hỗ trợ đa nền tảng, cho phép chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux.

-Với sự linh hoạt và mạnh mẽ của mình, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến và được ưa chuộng nhất trên thế giới, là lựa chọn hàng đầu cho cả học tập và phát triển công nghệ hiện đại.

2.MỘT SỐ MÔI TRƯỜNG PHỔ BIẾN

2.1.PyCharm

* PyCharm là một môi trường được phát triển để chuyên dành cho Python. PyCharm hỗ trợ nhiều tính năng giúp lập trình viên làm việc hiệu quả hơn, bao gồm:
* Hoàn thành mã thông minh: PyCharm cung cấp gợi ý hoàn thành mã, giúp lập trình viên viết mã nhanh hơn và ít lỗi hơn.
* Gỡ lỗi mạnh mẽ: Công cụ gỡ lỗi tích hợp của PyCharm rất mạnh mẽ, cho phép theo dõi và sửa lỗi dễ dàng.
* Hỗ trợ nhiều công cụ: PyCharm tích hợp nhiều công cụ như kiểm tra mã (code inspections), quản lý phiên bản (version control), và các công cụ phát triển web (Django, Flask).
* Giao diện thân thiện: PyCharm có giao diện đồ họa thân thiện và dễ sử dụng.

2.2.Jupyter Notebook

* Jupyter Notebook là một ứng dụng web mã nguồn mở cho phép tạo và chia sẻ các tài liệu có chứa mã nguồn, phương trình, hình ảnh và văn bản phong phú. Nó đặc biệt phổ biến trong cộng đồng khoa học dữ liệu và học máy. Một số tính năng chính của Jupyter Notebook bao gồm:
* Mã tương tác: Người dùng có thể viết và thực thi mã trong các ô (cell), cho phép kiểm tra và sửa mã nhanh chóng.
* Trực quan hóa dữ liệu: Jupyter hỗ trợ nhiều thư viện trực quan hóa dữ liệu như Matplotlib, Seaborn, và Plotly.
* Tài liệu sống: Kết quả đầu ra của mã (biểu đồ, bảng) được hiển thị ngay dưới ô mã, giúp dễ dàng theo dõi và phân tích kết quả.
* Hỗ trợ nhiều ngôn ngữ: Mặc dù chủ yếu được sử dụng cho Python, Jupyter cũng hỗ trợ nhiều ngôn ngữ khác như R, Julia, và Scala.

2.3.Visual Studio Code

* Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, nhẹ và mạnh mẽ được phát triển bởi Microsoft. VS Code đã trở thành một trong những trình soạn thảo mã phổ biến nhất, đặc biệt là cho Python. Một số tính năng nổi bật của VS Code bao gồm:
* Tiện ích mở rộng: VS Code có một kho tiện ích mở rộng phong phú, cho phép người dùng tùy chỉnh và mở rộng chức năng của trình soạn thảo.
* Hỗ trợ Python mạnh mẽ: Tiện ích mở rộng Python của VS Code cung cấp nhiều tính năng như tự động hoàn thành mã, kiểm tra lỗi, gỡ lỗi, và tích hợp Jupyter Notebook.
* Gỡ lỗi và điều hướng mã: VS Code có công cụ gỡ lỗi tích hợp và các tính năng điều hướng mã mạnh mẽ, giúp lập trình viên tìm và sửa lỗi nhanh chóng.
* Tích hợp Git: VS Code tích hợp Git, giúp quản lý phiên bản mã dễ dàng.

MỘT SỐ THƯ VIỆN PHỔ BIẾN

- Python có một thư viện tiêu chuẩn phong phú và một cộng đồng lớn mạnh, cung cấp hàng ngàn thư viện và module cho các ứng dụng khác nhau: từ phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, học máy, học sâu và tự động hóa hệ thống. Một số thư viện nổi tiếng của Python bao gồm NumPy, Pandas, Matplotlib, Scikit-learn....

NumPy

* NumPy (Numerical Python) là một thư viện chuyên về xử lý các mảng đa chiều và thực hiện các phép toán toán học. NumPy cung cấp một đối tượng mảng n-dimensional, gọi là ndarray, giúp thực hiện các phép toán số học nhanh chóng và hiệu quả. Ngoài ra, NumPy còn cung cấp các hàm toán học cấp cao để xử lý các mảng này.

Pandas

* Pandas là một thư viện mạnh mẽ cho việc phân tích và thao tác dữ liệu. Nó cung cấp hai cấu trúc dữ liệu chính là Series và DataFrame, giúp dễ dàng quản lý và thao tác với dữ liệu dạng bảng. Pandas cho phép đọc và ghi dữ liệu từ nhiều định dạng khác nhau (CSV, Excel, SQL, JSON), thực hiện các phép biến đổi dữ liệu phức tạp, xử lý dữ liệu bị thiếu, và nhiều thao tác khác, làm cho nó trở thành công cụ không thể thiếu trong phân tích dữ liệu và khoa học dữ liệu.

Matplotlib

* Matplotlib là một thư viện vẽ đồ thị 2D mạnh mẽ và linh hoạt trong Python. Nó cho phép tạo ra các biểu đồ, đồ thị, và hình ảnh trực quan từ dữ liệu. Matplotlib hỗ trợ nhiều kiểu đồ thị khác nhau như biểu đồ đường, biểu đồ thanh, biểu đồ phân tán, biểu đồ hộp, và nhiều kiểu khác. Thư viện này rất hữu ích cho việc trực quan hóa dữ liệu, giúp người dùng dễ dàng hiểu và phân tích dữ liệu một cách trực quan.

Scikit-learn

* Scikit-learn là một thư viện học máy (machine learning) phổ biến trong Python. Nó cung cấp các công cụ để thực hiện các thuật toán học máy như phân loại, hồi quy, clustering, và giảm chiều dữ liệu. Scikit-learn cũng cung cấp các công cụ để đánh giá mô hình và chọn mô hình, giúp dễ dàng xây dựng, huấn luyện, và đánh giá các mô hình học máy. Thư viện này rất phù hợp cho các ứng dụng học máy trong khoa học dữ liệu và trí tuệ nhân tạo.

PyQt6

* PyQt6 là một thư viện cho phép tạo ra các ứng dụng giao diện người dùng (GUI) bằng Python. Nó cung cấp các lớp và hàm để tạo ra các cửa sổ, hộp thoại, và các widget khác nhau. PyQt6 cho phép xây dựng các ứng dụng GUI phức tạp với nhiều tính năng và giao diện đẹp mắt. Thư viện này hỗ trợ đa nền tảng, có thể chạy trên Windows, macOS và Linux, làm cho nó trở thành lựa chọn phổ biến cho việc phát triển ứng dụng desktop bằng Python.

# **Chương 2.Tìm hiểu về thuật toán học máy**

## **1.Thuật toán Random Forest**

**Random Forest** là một trong những thuật toán học máy phổ biến được sử dụng cho các bài toán phân loại và hồi quy. Nó kết hợp nhiều cây quyết định (decision trees) để tạo ra một mô hình mạnh mẽ hơn và giảm thiểu các vấn đề như overfitting.

Trong dự đoán, thuật toán tổng hợp kết quả của tất cả các cây, bằng cách bỏ phiếu (đối với nhiệm vụ phân loại) hoặc bằng cách lấy trung bình (đối với nhiệm vụ hồi quy). Quá trình ra quyết định hợp tác này, được hỗ trợ bởi nhiều cây với thông tin chi tiết của chúng, cung cấp một ví dụ về kết quả ổn định và chính xác . Rừng ngẫu nhiên được sử dụng rộng rãi cho các chức năng phân loại và hồi quy, được biết đến với khả năng xử lý dữ liệu phức tạp, giảm tình trạng overfitting và cung cấp dự báo đáng tin cậy trong các môi trường khác nhau.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ

Mô tả được tạo tự động

**Các bước xây dựng mô hình Random Forest**

* **Bước 1: Lựa chọn số lượng cây (n\_estimators)**: Quyết định số lượng cây quyết định cần sử dụng trong mô hình. Số lượng cây càng nhiều thường dẫn đến hiệu suất dự đoán tốt hơn, nhưng đồng thời làm tăng thời gian huấn luyện và chi phí tính toán.
* **Bước 2: Huấn luyện các cây quyết định**: Mỗi cây quyết định được huấn luyện trên một tập dữ liệu con được lấy mẫu từ dữ liệu huấn luyện gốc, và chỉ sử dụng một số lượng ngẫu nhiên các đặc trưng.
* **Bước 3: Tổng hợp các dự đoán**: Dự đoán của Random Forest được tính toán bằng cách lấy trung bình (trong trường hợp hồi quy) hoặc bằng cách bỏ phiếu (trong trường hợp phân loại) của các cây quyết định thành viên.
* **Bước 4: Đánh giá và Tinh chỉnh mô hình**: Đánh giá hiệu suất của mô hình bằng các độ đo phù hợp như R-squared, Mean Absolute Error (MAE) hoặc Confusion Matrix (trong trường hợp phân loại). Có thể thực hiện tinh chỉnh mô hình bằng cách thay đổi các siêu tham số như số lượng cây, độ sâu của cây, hoặc số lượng đặc trưng sử dụng.

## **2.Transfer Learning**

**2.1. Khái niệm Transfer Learning**

**Transfer Learning** là một kỹ thuật trong lĩnh vực học máy, trong đó kiến thức được học từ một tác vụ hoặc một miền nguồn được chuyển giao và sử dụng để cải thiện hiệu suất của một tác vụ hoặc miền mục tiêu khác. Thay vì huấn luyện một mô hình từ đầu (từ ngẫu nhiên), ta sử dụng mô hình đã được huấn luyện trước đó trên một tác vụ liên quan để khởi tạo và tinh chỉnh lại cho tác vụ mới.

**2. Lý do sử dụng Transfer Learning**

* **Dữ liệu huấn luyện ít**: Trong nhiều trường hợp thực tế, có ít dữ liệu huấn luyện khả dụng để huấn luyện mô hình từ đầu. Transfer Learning giúp tận dụng các tri thức từ dữ liệu lớn của các tác vụ khác để cải thiện hiệu suất mô hình.
* **Tăng tốc quá trình huấn luyện**: Việc sử dụng các mô hình đã được huấn luyện trước đó giúp tiết kiệm thời gian và công sức so với việc huấn luyện từ đầu. Các mô hình đã có sẵn thường có thể học các đặc trưng chung và trừu tượng của dữ liệu một cách hiệu quả.
* **Giải quyết vấn đề Overfitting**: Các mô hình đã được huấn luyện trước đó thường có khả năng tổng quát hóa tốt hơn do đã được huấn luyện trên nhiều dữ liệu. Việc sử dụng Transfer Learning có thể giúp giảm thiểu hiện tượng overfitting, đặc biệt khi dữ liệu huấn luyện ít.

**3. Các Bước Thực Hiện Transfer Learning**

* **Chọn mô hình cơ sở (Base Model)**: Lựa chọn một mô hình đã được huấn luyện trước đó, thường là trên một bộ dữ liệu lớn như ImageNet, để sử dụng làm mô hình cơ sở.
* **Dong băng các lớp của mô hình cơ sở**: Đóng băng (freeze) các lớp của mô hình cơ sở để giữ nguyên các trọng số đã học được từ tác vụ gốc.
* **Thêm các lớp mới (Optional)**: Thêm một hoặc nhiều lớp phù hợp với tác vụ mới, thường là các lớp Fully Connected Layer, để phù hợp với số lượng lớp đầu ra của tác vụ mới.
* **Tinh chỉnh mô hình (Fine-tuning)**: Tùy chỉnh lại các trọng số của mô hình bằng cách huấn luyện lại một số lớp trên dữ liệu mới của tác vụ, thường là với learning rate thấp để tránh làm mất đi các đặc trưng đã học được.

**4. Lợi ích của Transfer Learning**

* **Cải thiện hiệu suất**: Giúp cải thiện hiệu suất của mô hình trên tác vụ mới mà không cần nhiều dữ liệu huấn luyện.
* **Tiết kiệm thời gian và tài nguyên**: Giảm thiểu thời gian và tài nguyên cần thiết để huấn luyện mô hình từ đầu.
* **Ứng dụng rộng rãi**: Áp dụng được cho nhiều lĩnh vực và các tác vụ khác nhau trong học máy và thị giác máy tính.

# **Chương 3.Bài toán dự đoán giá vàng**

Các bước thực hiện.

## **Bước 1: Thu Thập Dữ Liệu**

* Dữ liệu lịch sử giá vàng được lấy từ trang web Kaggle . <https://www.kaggle.com/datasets/altruistdelhite04/gold-price-data>
* Dữ liệu gồm 6 cột và 2280 dòng

| **Date** | **SPX** | **GLD** | **USO** | **SLV** | **EUR/USD** |
| --- | --- | --- | --- | --- | --- |
| 1/2/2008 | 1447.160034 | 84.860001 | 78.470001 | 15.180 | 1.471692 |
| 1/3/2008 | 1447.160034 | 85.570000 | 78.370003 | 15.285 | 1.474491 |
| 1/4/2008 | 1411.630005 | 85.129997 | 77.309998 | 15.167 | 1.475492 |
| 1/7/2008 | 1416.180054 | 84.769997 | 75.500000 | 15.053 | 1.468299 |
| 1/8/2008 | 1390.189941 | 86.779999 | 76.059998 | 15.590 | 1.557099 |
|  |  |  |  |  |  |

**Các cột:**

1. **SPX:** Giá trị chỉ số S&P 500.
2. **GLD:** Giá vàng tính bằng USD trên mỗi ounce.
3. **USO:** Giá của Quỹ Dầu Mỹ (United States Oil Fund) tính bằng USD.
4. **SLV:** Giá bạc tính bằng USD trên mỗi ounce.
5. **EUR/USD:** Tỷ giá hối đoái của Euro so với Đô la Mỹ

Thư viện cần thêm vào :Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

## **Bước 2: Khám Phá và Xử Lý Dữ Liệu**

1.Đọc dữ liệu

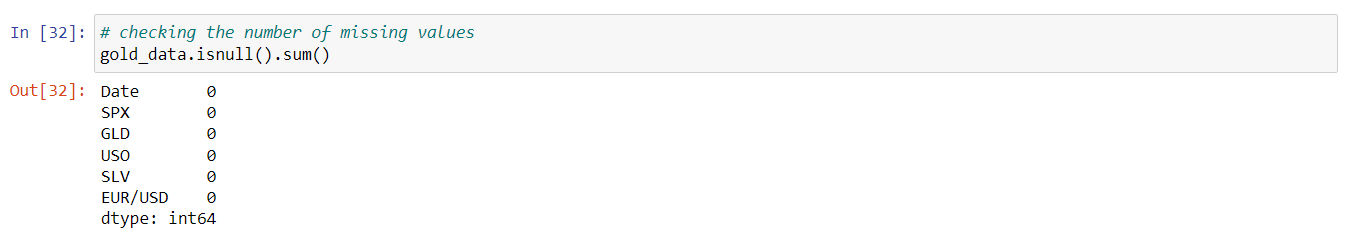


2.Hiện thị thông tin cơ bản của dữ liệu

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

3.Kiểm tra giá trị thiếu



Không có giá trị thiếu

4.Tổng hợp số liệu thống kê

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

-**count (Số lượng):** Số lượng mẫu dữ liệu được sử dụng để tính toán các thông số thống kê. Ở đây, tất cả các cột đều có 2290 mẫu.

-**mean (Giá trị trung bình):** Giá trị trung bình của các mẫu dữ liệu.

-**std (Độ lệch chuẩn):** Đo lường mức độ phân tán của các mẫu dữ liệu so với giá trị trung bình.

- **min (Giá trị nhỏ nhất):** Giá trị nhỏ nhất trong các mẫu dữ liệu.

- **25% (Phân vị thứ 25):** Giá trị mà 25% các mẫu dữ liệu nhỏ hơn hoặc bằng giá trị này (còn gọi là Q1 hoặc quartile thứ nhất).

- **50% (Phân vị thứ 50 hoặc Median):** Giá trị trung vị, là giá trị nằm giữa các mẫu dữ liệu (còn gọi là median).

-**75% (Phân vị thứ 75):** Giá trị mà 75% các mẫu dữ liệu nhỏ hơn hoặc bằng giá trị này (còn gọi là Q3 hoặc quartile thứ ba).

- **max (Giá trị lớn nhất):** Giá trị lớn nhất trong các mẫu dữ liệu.

## **Bước 3: Phân Tích Dữ Liệu**

1.Tính toán ma trận tương quan



2.Vẽ heatmap để hiện thị ma trận tương quan

Ảnh có chứa ảnh chụp màn hình, văn bản, hình vuông, biểu đồ

Mô tả được tạo tự động

## **Bước 4: Chuẩn bị dữ liệu**

1.Xác định features và target:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động Ảnh có chứa văn bản, đồ điện tử, ảnh chụp màn hình, màn hình

Mô tả được tạo tự động

2.Chia dữ liệu thành tập training và tập test



Test\_size = 0.2 nghĩa là 20% dữ liệu sẽ được phân bổ vào tập kiểm thử. Tỷ lệ còn lại (80%) sẽ thuộc về tập huấn luyện

Random\_state :ham số này dùng để thiết lập seed cho hàm random, giúp đảm bảo tính khả tái tạo (reproducibility) của việc phân chia dữ liệu

## **Bước 5: Xây dựng mô hình**

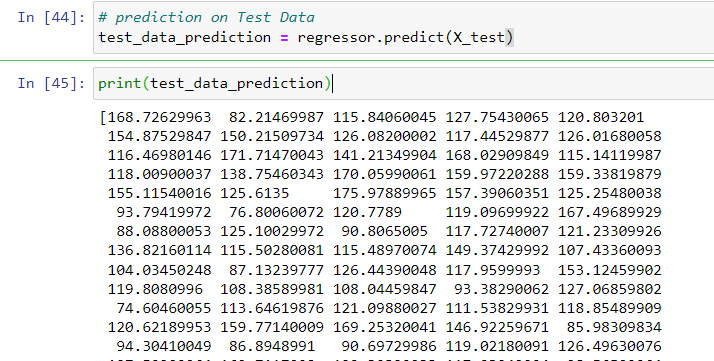
1.Khởi tạo và huấn luyện mô hình Random Forest

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

## **Bước 6: Đánh giá mô hình**

1.Dự đoán trên tập kiểm tra:



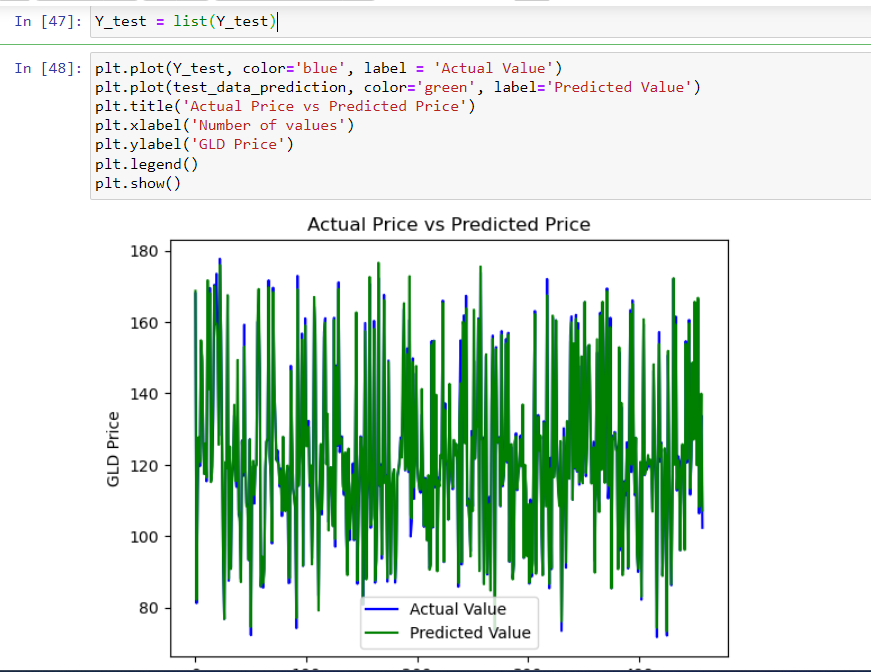
2.Tính toán lỗi R-squared

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Mô hình đúng trên tập test 98,85%

## **Bước 7: Trực quan hóa kết quả**



## **Bước 8: Triển khai**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

# **Chương 4.Bài toán nhận diện cảm xúc**

Các bước thực hiện :

## **Bước 1: Thu Thập Dữ Liệu và đọc ảnh**

-Dữ liệu được lấy từ trang web Kaggle gồm các hình ảnh đã được phân loại

<https://www.kaggle.com/datasets/msambare/fer2013>

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

**Đọc và hiển thị ảnh**: Sử dụng thư viện OpenCV để đọc ảnh từ thư mục chứa dữ liệu huấn luyện. Hiển thị ảnh bằng Matplotlib.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

**Tạo danh sách lớp**: Tạo danh sách các lớp cảm xúc (angry ,disgust,fear, happy,neutral,sad,supriesed tương ứng "0", "1", "2", "3", "4", "5", "6" bằng cách đổi tên thư mục).

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

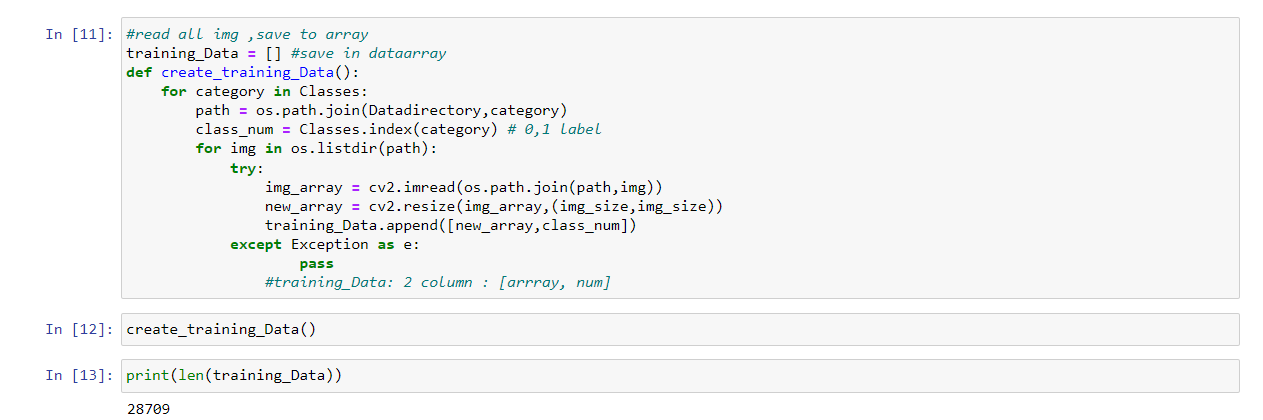
**Thay đổi kích thước ảnh**: Tất cả các ảnh được thay đổi kích thước thành 224x224 để phù hợp với mô hình MobileNetV2.

Ảnh có chứa văn bản, Mặt người, em bé, ảnh chụp màn hình

Mô tả được tạo tự động

## **Bước 2.Tạo dữ liệu huấn luyên**

**1.Tạo mảng dữ liệu huấn luyện**: Duyệt qua từng lớp, đọc từng ảnh, thay đổi kích thước và lưu vào mảng dữ liệu huấn luyện.



**2.Shuffle dữ liệu**: Sử dụng hàm random.shuffle để trộn lẫn dữ liệu, giúp cải thiện quá trình huấn luyện.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

**3.Tách đặc trưng và nhãn**: Tách mảng dữ liệu huấn luyện thành hai phần: X (dữ liệu ảnh) và y (nhãn tương ứng).

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màn hình

Mô tả được tạo tự động

## **Bước 3: Chuẩn bị mô hình**

**1.Chuyển đổi và chuẩn hóa dữ liệu**: Chuyển đổi X thành dạng numpy array và chuẩn hóa pixel

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

**2.Sử dụng mô hình MobileNetV2**: Tải mô hình MobileNetV2 đã được huấn luyện trước và sử dụng làm mô hình cơ sở.



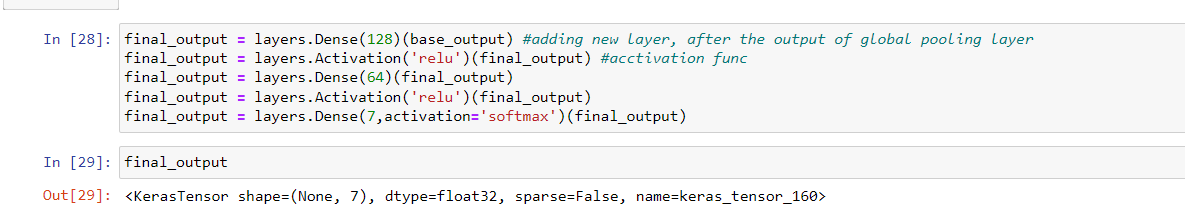
**3.Đóng băng các lớp của mô hình cơ sở**: Để không cho phép thay đổi trọng số của các lớp này trong quá trình huấn luyện.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

## **Bước 4: Xây dựng mô hình mới**

**1.Thêm các lớp mới**: Thêm các lớp Dense và Activation vào mô hình để phù hợp với bài toán nhận diện cảm xúc (7 lớp đầu ra).



**2.Tạo mô hình hoàn chỉnh**: Tạo mô hình mới từ đầu vào của MobileNetV2 và đầu ra từ lớp Dense cuối cùng.

Ảnh có chứa ảnh chụp màn hình, văn bản, hàng, Phông chữ

Mô tả được tạo tự động

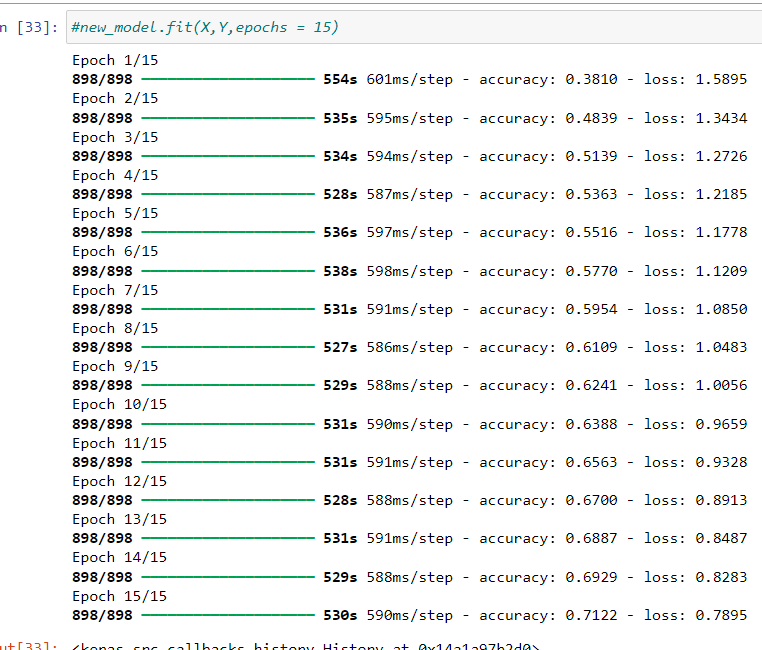
**3.Compile mô hình**: Sử dụng hàm mất mát "sparse\_categorical\_crossentropy", bộ tối ưu hóa "adam", và đo lường độ chính xác.



## **Bước 5: Huấn luyện mô hình**

**1.Huấn luyện**: Huấn luyện mô hình trên dữ liệu X và Y với 15 epoch.

càng nhiều epoch, mô hình càng chính xác, ở đây với epoch 15, độ chính xác là 71%



**2.Lưu mô hình**: Lưu mô hình đã được huấn luyện để sử dụng sau này.



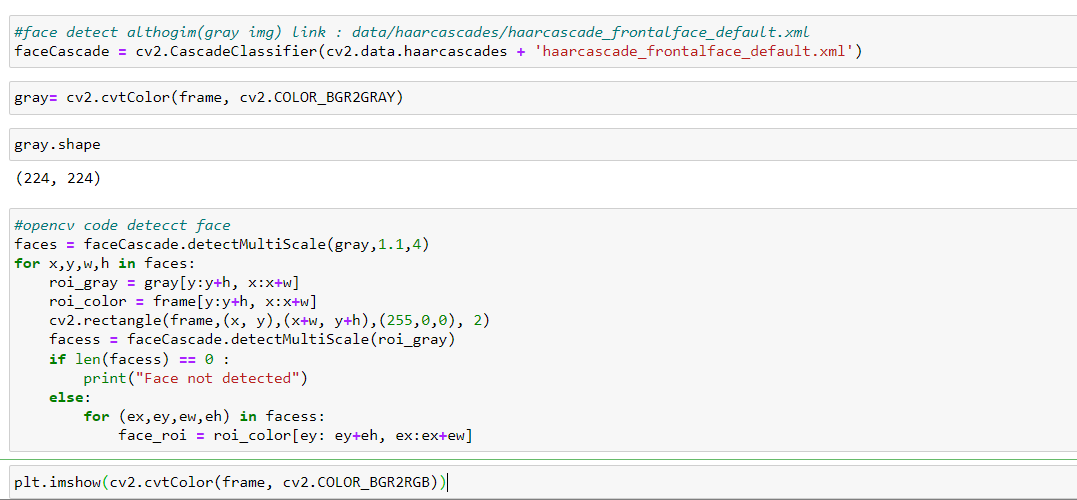
## **Bước 6: Nhận diện cảm xúc từ ảnh mới**

**1.Đọc ảnh**: Đọc ảnh mới bằng OpenCV.

Ảnh có chứa văn bản, Mặt người, ảnh chụp màn hình, người

Mô tả được tạo tự động

**2**.**Phát hiện khuôn mặt**: Sử dụng Haar Cascade để phát hiện khuôn mặt trong ảnh.



Ảnh có chứa Mặt người, ảnh chụp màn hình, văn bản, da

Mô tả được tạo tự động

**3.Cắt và chuẩn bị ảnh khuôn mặt**: Cắt phần khuôn mặt từ ảnh gốc và thay đổi kích thước thành 224x224. Chuẩn hóa giá trị pixel.

Ảnh có chứa văn bản, Mặt người, da, ảnh chụp màn hình

Mô tả được tạo tự động Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

**4.Dự đoán cảm xúc**: Sử dụng mô hình đã huấn luyện để dự đoán cảm xúc từ ảnh khuôn mặt. Hiển thị kết quả dự đoán.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

* + - Kết quả của khuôn mặt dự đoán: neutral : chính xác

# **Chương 5.Kết luận**

-Mô hình Random Forest đã giúp dự đoán giá vàng với độ chính xác cao. Qua các bước từ xử lý dữ liệu, xây dựng mô hình đến đánh giá và trực quan hóa kết quả, mô hình đã cho thấy khả năng dự đoán hiệu quả.

-Mô hình nhận diện cảm xúc khuôn mặt đã dự đoán khá chính xác, tuy nhiên qua kiểm thử nhiều ảnh, nó vẫn nhầm lẫn một số ảnh có cảm xúc khó.

Ví dụ: cảm xúc của khuôn mặt này là sad, nhưng khi test thì trả về kết quả là happy



-Hướng phát triển trong tương lai: Để cải thiện mô hình nhận diện cảm xúc khuôn mặt, tương lai có thể sử dụng nhiều data hơn, training với hệ số epoch nhiều hơn để có độ chính xác cao nhất có thể, tuy nhiên sẽ tốn khá nhiều thời gian và cần cấu hình máy khỏe hơn để huấn luyện

# **Tài liệu tham khảo**

1.[Random Forest tutorial](https://www.ibm.com/topics/random-forest%23:~:text=Random%20forest%20is%20a%20commonly,Decision%20trees)

2.[MobileNetV2 explain](https://paperswithcode.com/method/mobilenetv2)

3.[Github tham khảo emotion recognition](https://github.com/spignelon/InnerView)