**Bài tập thực hành 06:**

Save/Load model và triển khai

# **Mục tiêu cần đạt**:

* Biết cách lưu model đã được huấn luyện
* Biết cách load model để áp dụng
* Phân biệt các kiểu lưu mô hình máy học
* Biết cách triển khai để áp dụng để áp dụng vào 1 bài toán phân lớp/hồi quy

# Một số tài liệu tham khảo

<https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/>

<https://machinelearningmastery.com/save-load-keras-deep-learning-models/>

<https://www.kaggle.com/prmohanty/python-how-to-save-and-load-ml-models>

# Các bước thực hiện việc lưu trữ mô hình đã huấn luyện

Trong bài hướng dẫn các bạn sẽ được học cách lưu trữ các mô hình Máy học cổ điển scikit-learn và các thuật toán học sâu với keras. Với mỗi lần lưu trữ chúng ta sẽ trải qua các bước

* Load dữ liệu để huấn luyện
* Định nghĩa kiến trúc/khai báo các siêu tham số sử dụng/khởi tạo mô hình
* Cho mô hình khởi tạo học dữ liệu đã được load
* Lưu mô hình đã được huấn luyện vào ổ đĩa
* Triển khai

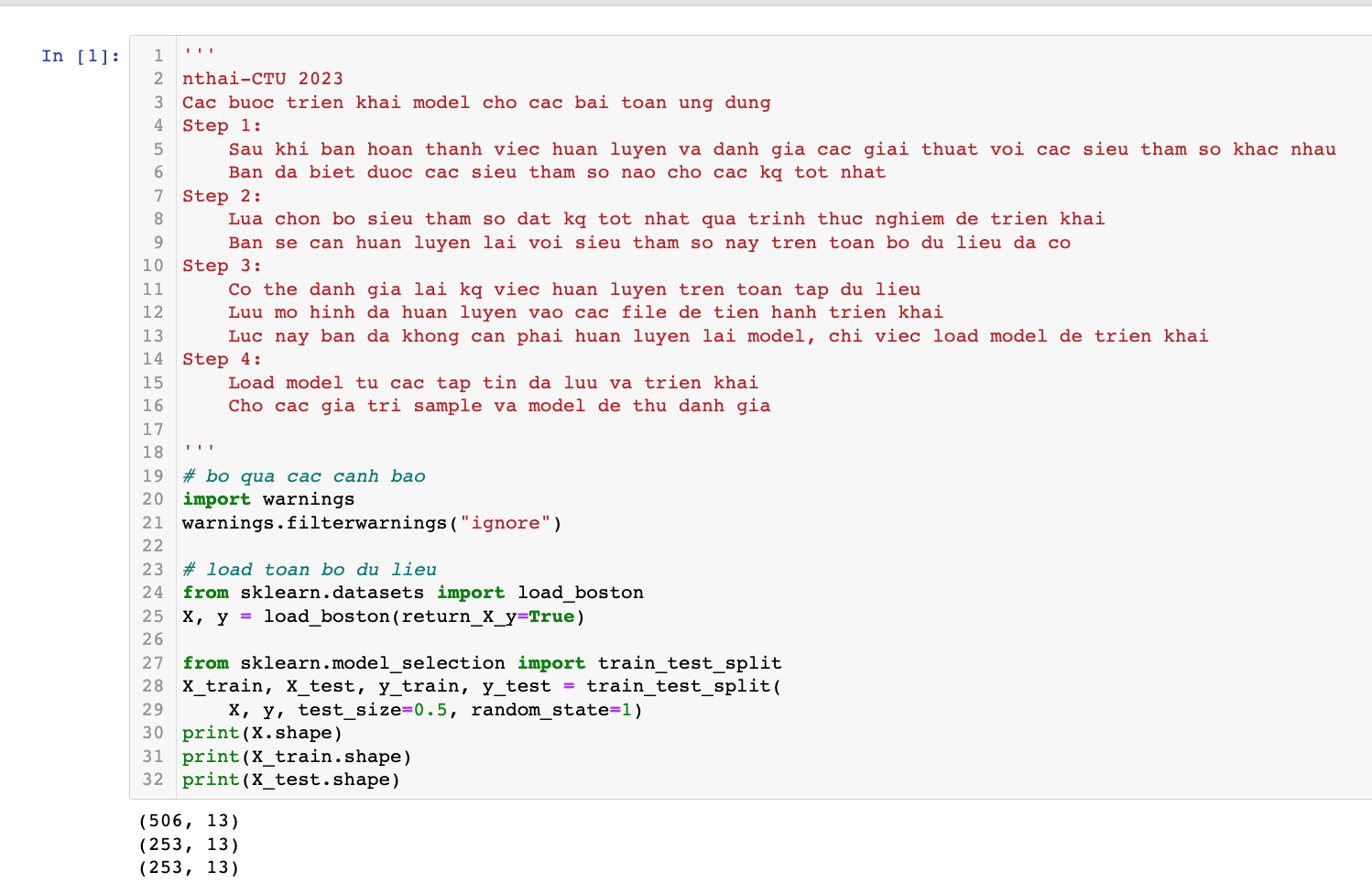
Sau này khi dữ liệu có nhiều hơn để cải thiện độ chính xác, các bạn có thể load lại mô hình và huấn luyện trên tập dữ liệu mới đã được bổ sung

Trong nội dung bài thực hành này sẽ đưa ra các ví dụ để lưu các mô hình máy học cổ điển từ sklearn, các mô hình mạng nơ-ron tích chập từ keras với 2 bài toán hồi quy và phân lớp.

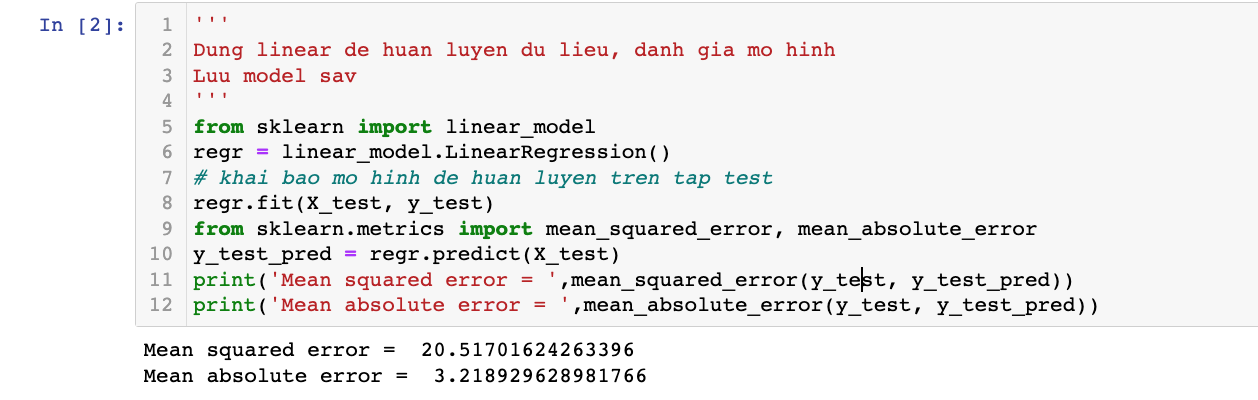
## Lưu trữ mô hình máy học cổ điển

### Chúng ta load dữ liệu:

Trong ví dụ này chúng ta sẽ thử nghiệm với dữ liệu giá nhà. Ta thử chia dữ liệu ra 2 phần Train/Test và chúng ta thử cho huấn luyện trên dữ liệu Test và tập chưa chia để xem các mô hình dự đoán sẽ có kết quả thế nào



Đánh giá độ lỗi trong quá trình huấn luyện

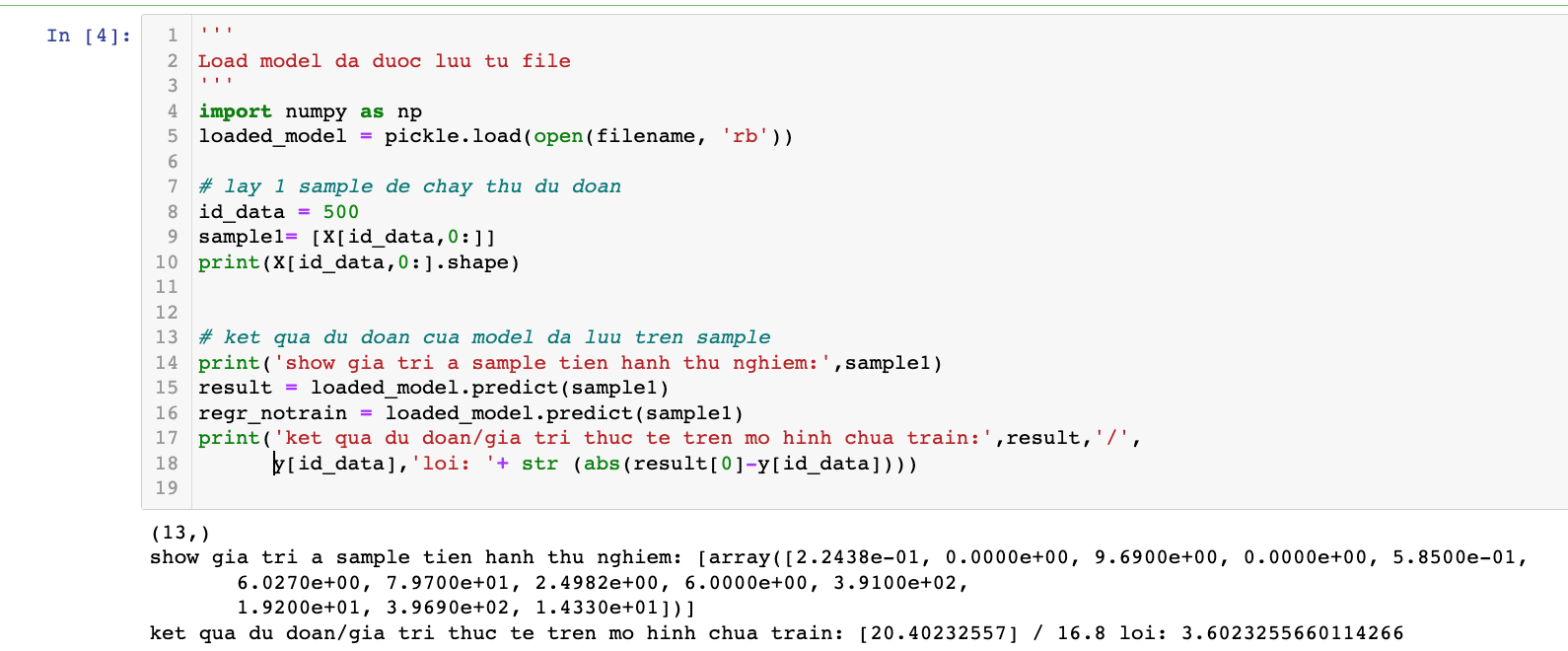


### Lưu mô hình đã huấn luyện

Triển khai mô hình đã huấn luyện load từ file đã lưu và thử lấy mẫu thứ 500 trong tập dữ liệu chưa chia để thực hiện dự đoán, chúng ta thấy được độ lỗi

### Load mô hình đã huấn luyện

Kết quả dự đoán trên mô hình huấn luyện trên tập kiểm tra



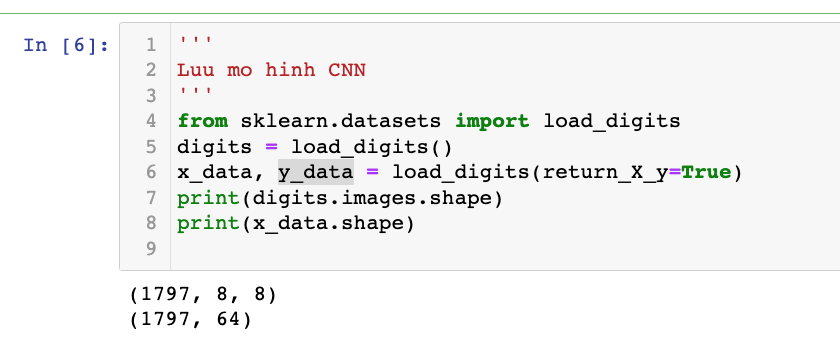
Kết quả dự đoán trên mô hình huấn luyện trên toàn tập chưa chia

## 

## Lưu trữ mô hình học sâu

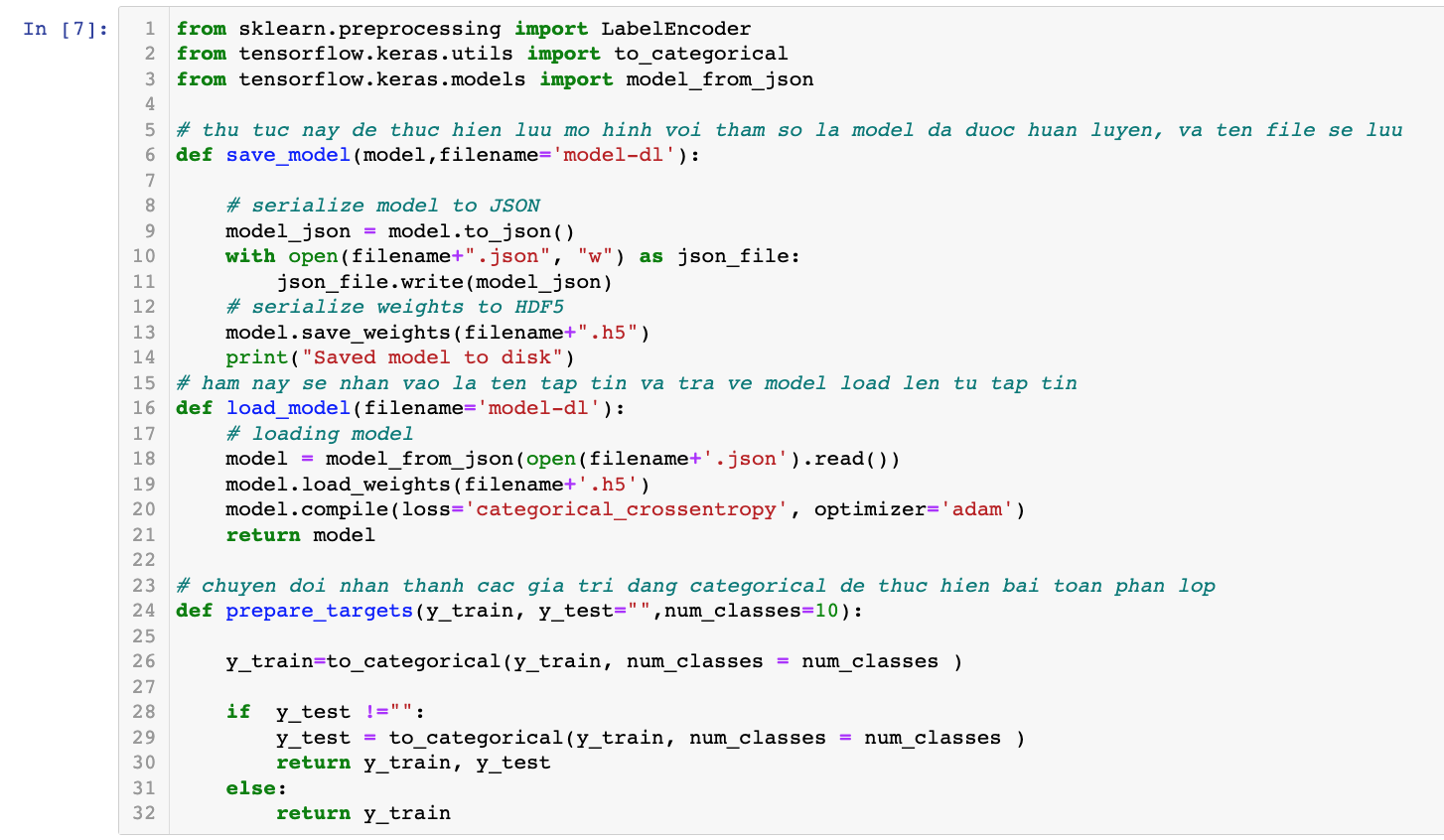
Chúng ta cũng thực hiện tương tự, với khối lệnh [6] chúng ta sẽ

### Load dữ liệu lên



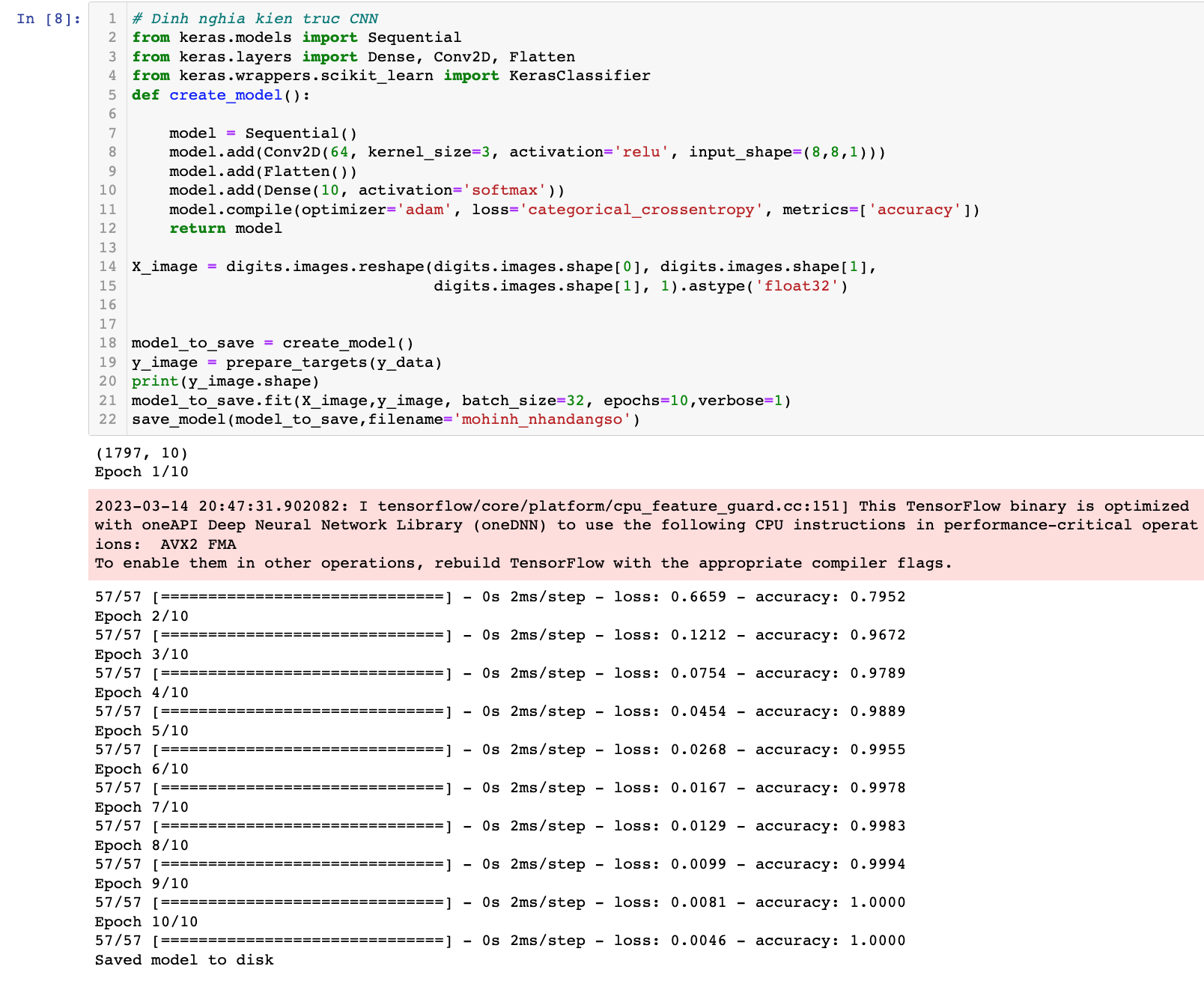
### Định nghĩa 1 số hàm/thủ tục hỗ trợ: Khối lệnh [7], chúng ta định nghĩa các hàm/thủ tục để lưu/load mô hình, với mô hình deep learning từ thư viện keras chúng ta sẽ có 2 file là .h5 (lưu các trọng số) và .json (thông tin các layer, kiến trúc)

Trong khối lệnh này hỗ trợ chuyển đổi nhãn sang dạng category cho bài toán phân lớp đa lớp.



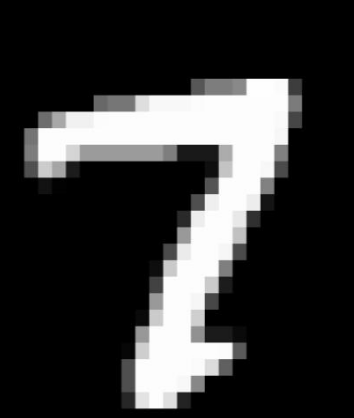
### Khởi tạo và huấn luyện: Ở khối lệnh [8], chúng ta sẽ định nghĩa kiến trúc CNN và bắt đầu huấn luyện.

Sau khi huấn luyện xong, chúng ta sẽ lưu lại mô hình vào các file h5 và json



### Triển khai mô hình đã huấn luyện trên các ảnh thu thập

*(Lưu ý: để có 2 hình so07.png và so0.png để thực hiện việc dự đoán bạn có thể dùng phần mềm Snipping Tool trong Windows để chụp lại 2 ảnh bên dưới và lưu thành so07.png và so0.png đặt cùng thư mục với file jupyter, hoặc các bạn thử download ảnh khác từ trên mạng để trải nghiệm)*

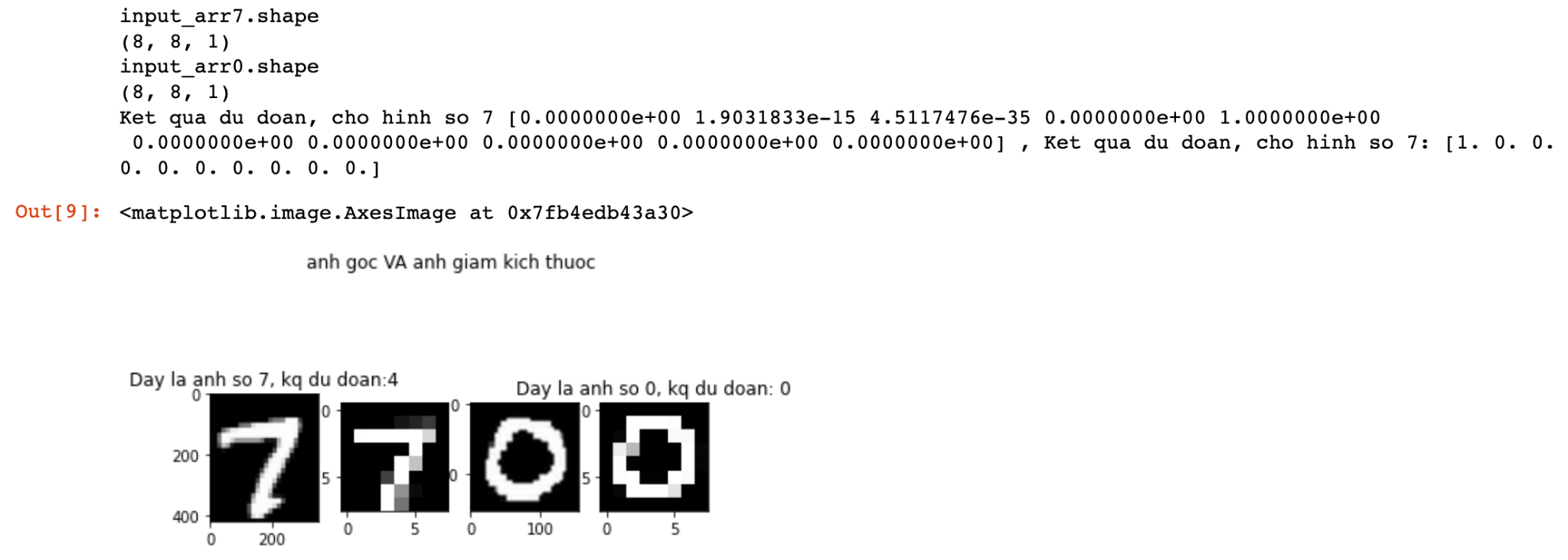




Khối lệnh [9], chúng ta có 2 hình ảnh số 7 và số 0 để thử thực hiện dự đoán trên ảnh này. Trước tiên chúng ta load ảnh bao gồm ảnh gốc và ảnh được scale lại để vừa với mô hình vốn chỉ huấn luyện trên tập ảnh xám kích thước 8x8. Chúng ta thêm dòng 12, 13 để chuyển sang array thuận tiện cho việc đưa vào mô hình thực hiện dự đoán.



Quan sát khối lệnh từ 16-19 thể hiện kích thước của các ảnh đưa vào phù hợp với input\_shape ở khối lệnh [8] ta có thể đưa vào mô hình để dự đoán với phương thức *predict* ở dòng 28.

Kết quả dòng 29 cho chúng ta 1 mảng 10 phân tử cho mỗi ảnh tương ứng với xác suất mà ảnh này thuộc về ở 10 phân lớp số từ 0-9.

HÌnh số 7 thuộc về phân lớp 4 với xác suất 1.0000e+00, như vậy là mô hình đã nhận dạng nhầm hình này là số 4, nhưng nó cũng có 1 chút dự đoán là số 1 với xác suất 1.9e-15, và nhận nhầm là số 2 với xác suất là 4.5e-35.

Với số 0 thì mô hình dễ dàng nhận ra với xác suất 1, trong khi các phân lớp còn lại đều là 0.

# Yêu cầu bài thực hành:

* *Với file word: Với các câu hỏi về lệnh các bạn trả lời bằng text: copy lệnh (dạng text) vào bài word trả lời và kèm hình ảnh minh họa. Tất cả câu trả lời phải có hình minh họa minh chứng kết quả*
* *Với lệnh trong Jupyter: ở đầu mỗi cell các bạn tạo ghi chú để rõ phần trả lời của câu nào*

1. Bạn hãy liệt kê các cách lưu trữ các mô hình đã được huấn luyện, phân tích ưu và khuyết điểm mỗi cách, bạn có thể tham khảo từ[[1]](#footnote-1)[[2]](#footnote-2)[[3]](#footnote-3)

***Bài làm***

Có nhiều cách để lưu trữ các mô hình đã được huấn luyện trong machine learning, tùy thuộc vào loại mô hình, thư viện và nhu cầu sử dụng. Dưới đây là một số cách phổ biến và ưu nhược điểm của chúng:

* Lưu trữ mô hình dưới dạng file nhị phân: Đây là cách đơn giản nhất để lưu trữ một mô hình, chỉ cần dùng các hàm có sẵn của thư viện để ghi và đọc mô hình vào một file nhị phân. Ví dụ, trong scikit-learn, có thể dùng pickle hoặc joblib để lưu và tải mô hình. Ưu điểm của cách này là nhanh chóng, dễ dàng và không cần phải quan tâm đến kiến trúc hay trọng số của mô hình. Nhược điểm là file nhị phân không thể đọc được bằng con người, không tương thích với các phiên bản khác nhau của thư viện hoặc các ngôn ngữ khác nhau.
* Lưu trữ mô hình dưới dạng file văn bản: Đây là cách lưu trữ một mô hình bằng cách ghi các thông tin về kiến trúc, trọng số, siêu tham số và các thông số khác của mô hình vào một file văn bản có định dạng nhất định. Ví dụ, trong Keras, có thể dùng JSON hoặc YAML để lưu kiến trúc của mô hình và HDF5 để lưu trọng số của mô hình. Ưu điểm của cách này là file văn bản có thể đọc được bằng con người, có thể chỉnh sửa hoặc chuyển đổi sang các định dạng khác. Nhược điểm là cần phải lưu nhiều file khác nhau cho một mô hình, không thể lưu được các đối tượng phức tạp như các lớp tự định nghĩa hay các hàm tùy biến.
* Lưu trữ mô hình dưới dạng mã nguồn: Đây là cách lưu trữ một mô hình bằng cách viết lại toàn bộ mã nguồn để xây dựng và huấn luyện mô hình. Ví dụ, trong PyTorch, có thể dùng torch.save() để lưu toàn bộ mã nguồn của một lớp hay một hàm vào một file. Ưu điểm của cách này là có thể tái sử dụng hoặc sửa đổi mã nguồn cho các mục đích khác nhau, không bị phụ thuộc vào phiên bản của thư viện hay ngôn ngữ. Nhược điểm là cần phải viết lại mã nguồn cho từng loại mô hình, không thể lưu được các thông tin khác như kết quả huấn luyện hay độ chính xác của mô hình.

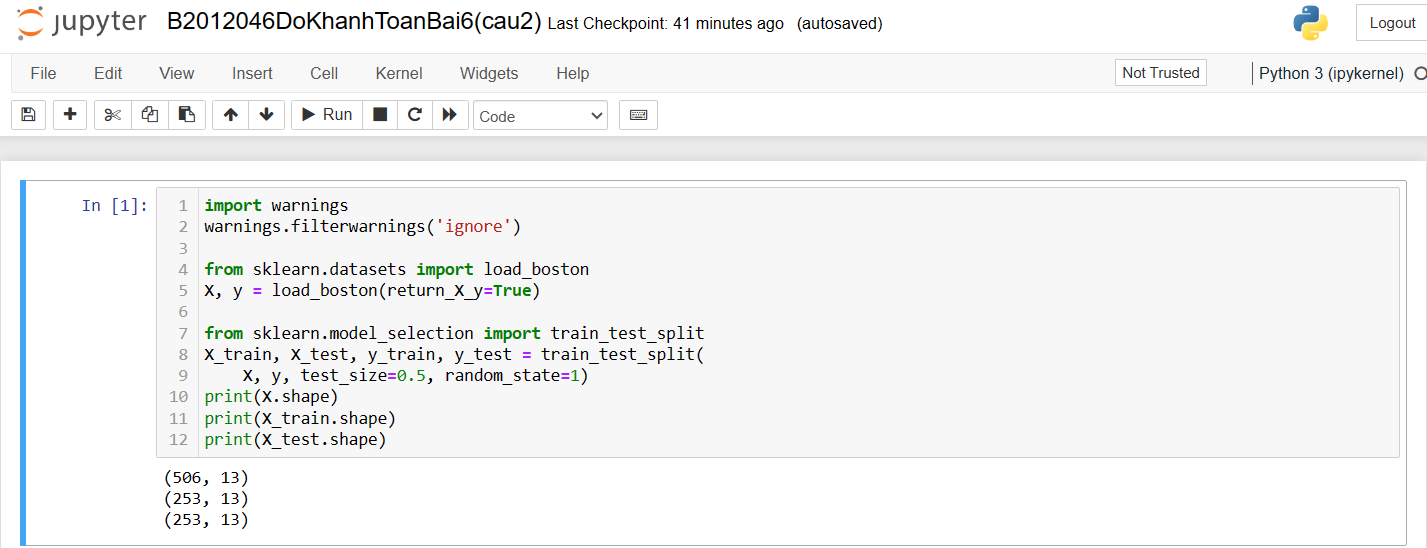
Link tham khảo:

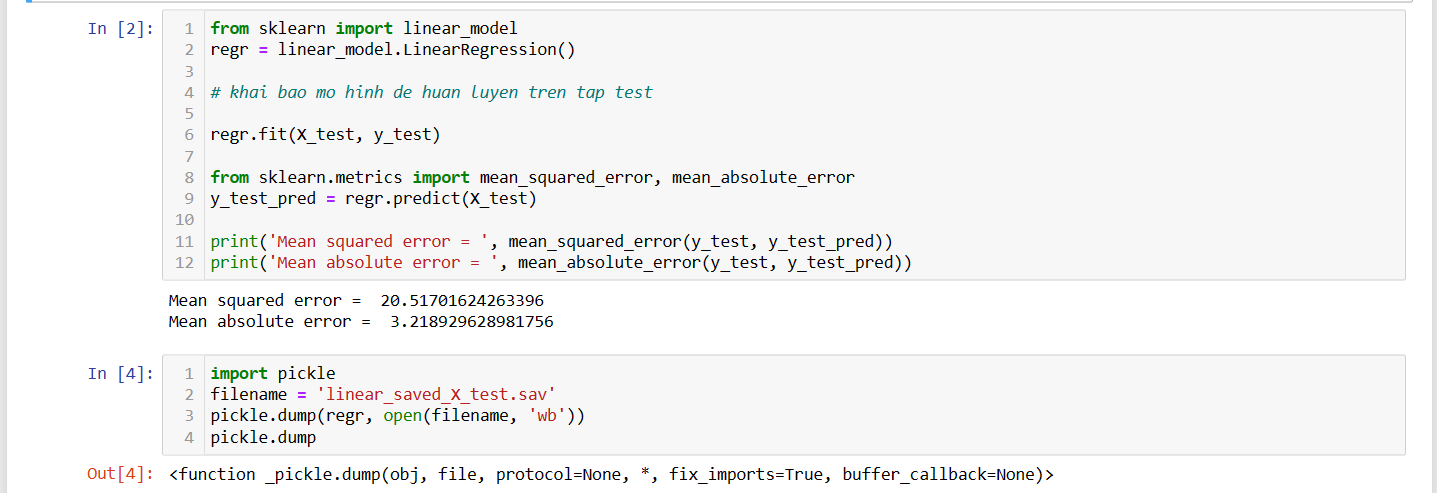
<https://www.ods.vn/tin-cong-nghe/luu-tru-du-lieu-la-gi.html>

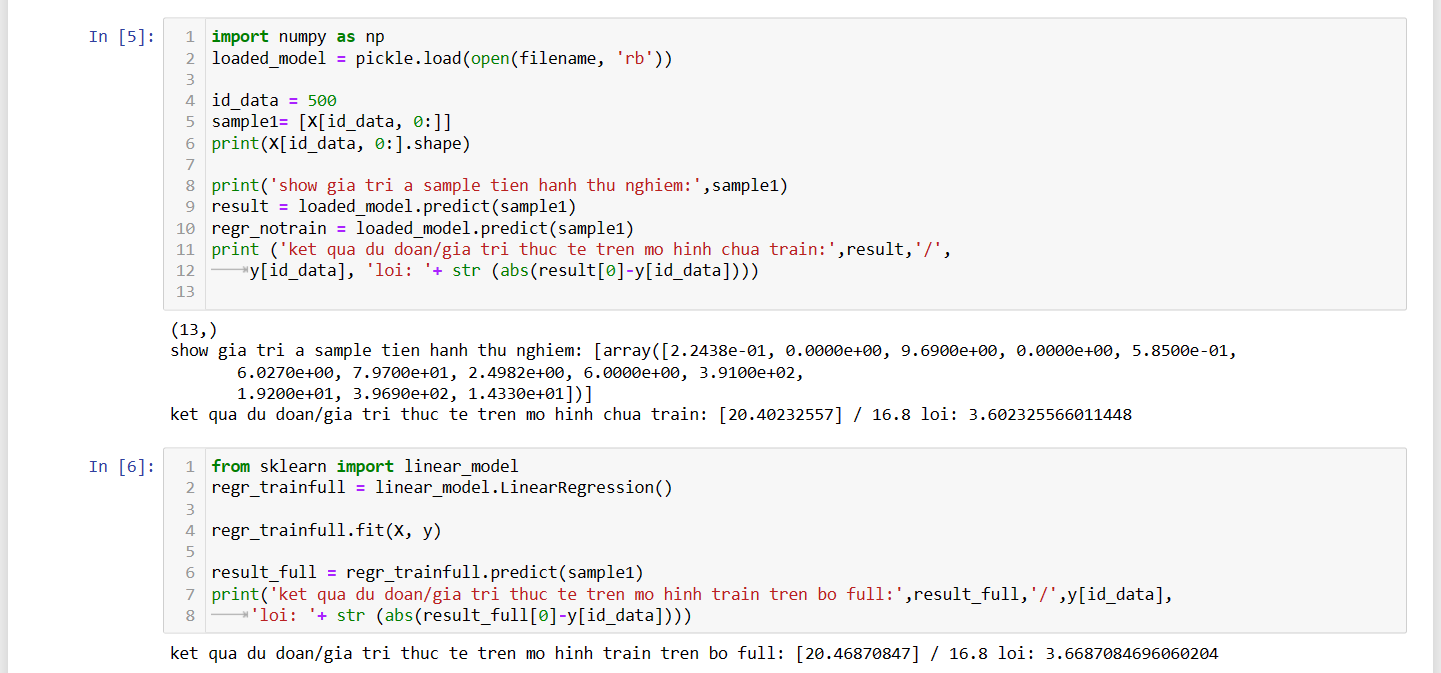
<https://itguru.vn/blog/tong-hop-nhung-nguon-datasets-danh-cho-machine-learning/>

1. Bạn hãy thực hiện lại bài thực hành theo hướng dẫn ở trên.

***Bài làm***

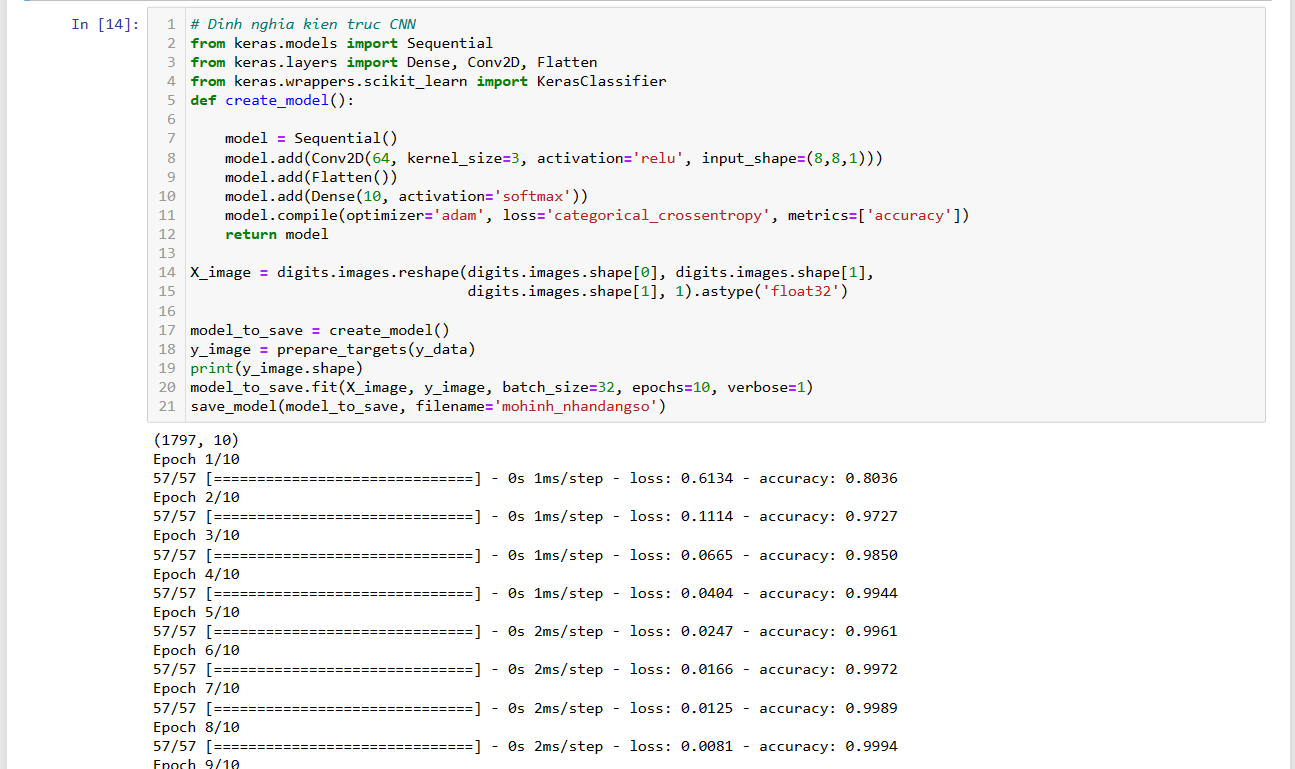
******

******

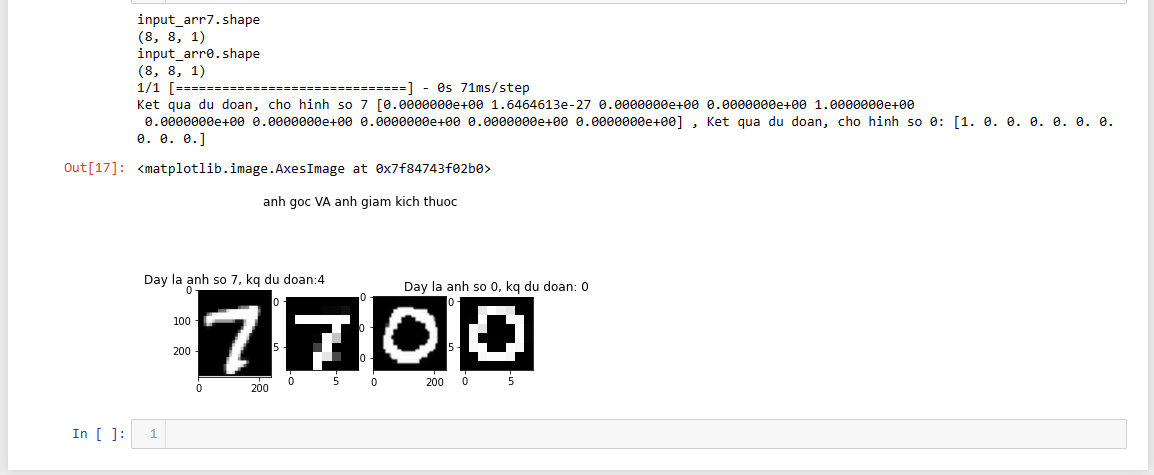
******

******

******

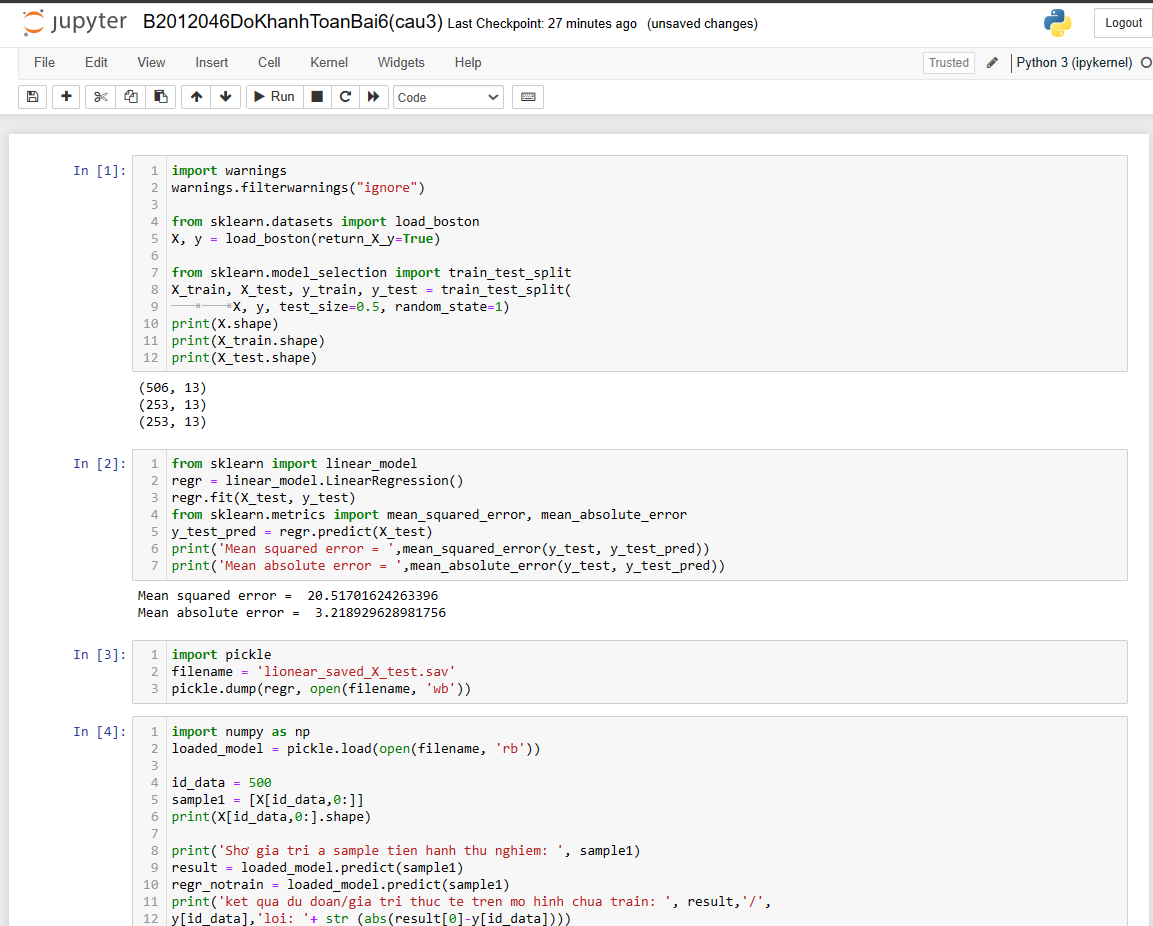
******

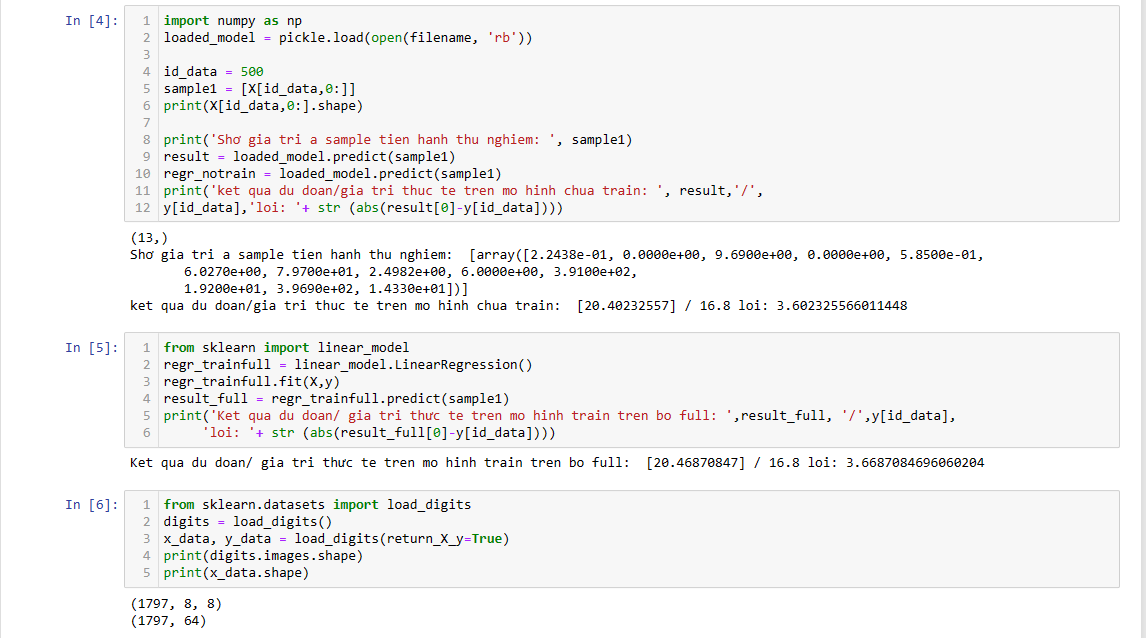
******

******

1. Vào thư mục chứa file jupyter sau khi bạn chạy các lệnh hướng dẫn ở trên, thử xóa file **.json** đã được tạo ra sau quá trình huấn luyện, và load thử mô hình xem việc gì sẽ diễn ra, thử giải thích tại sao.

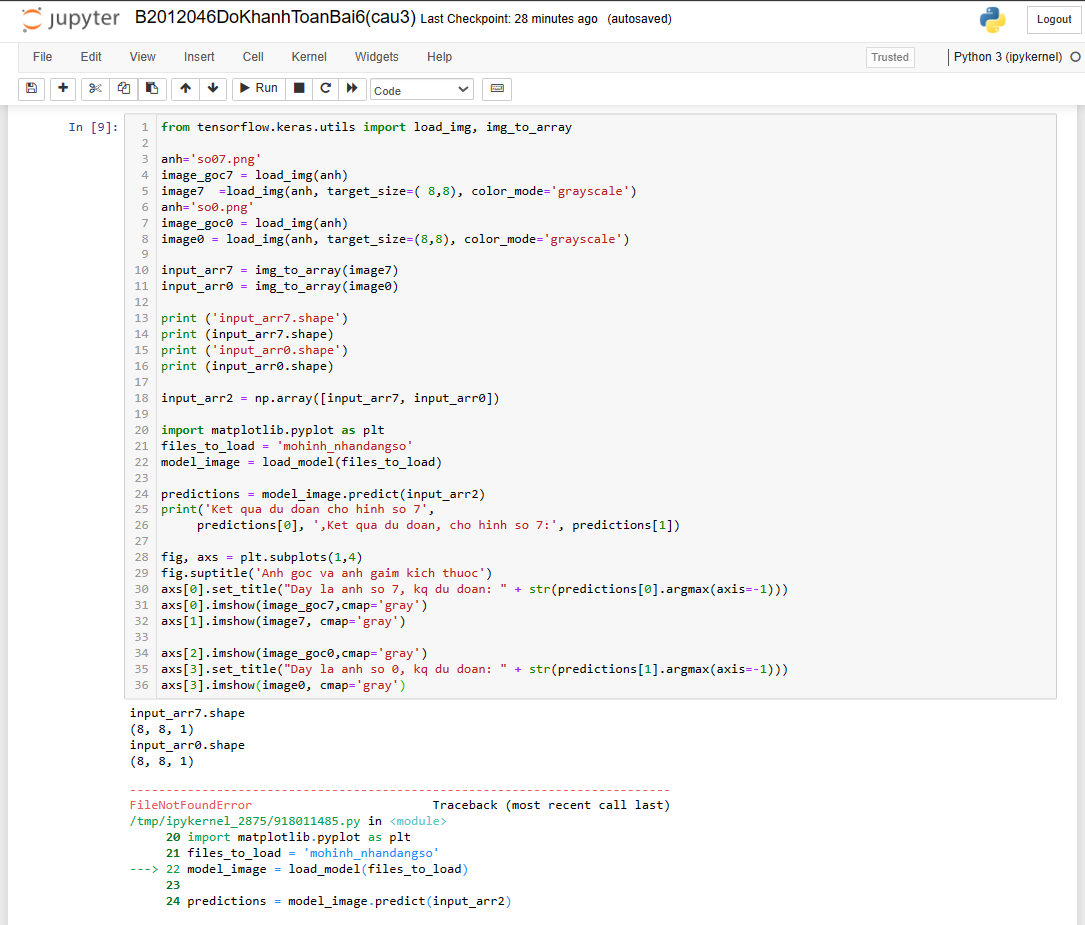
***Bài làm***

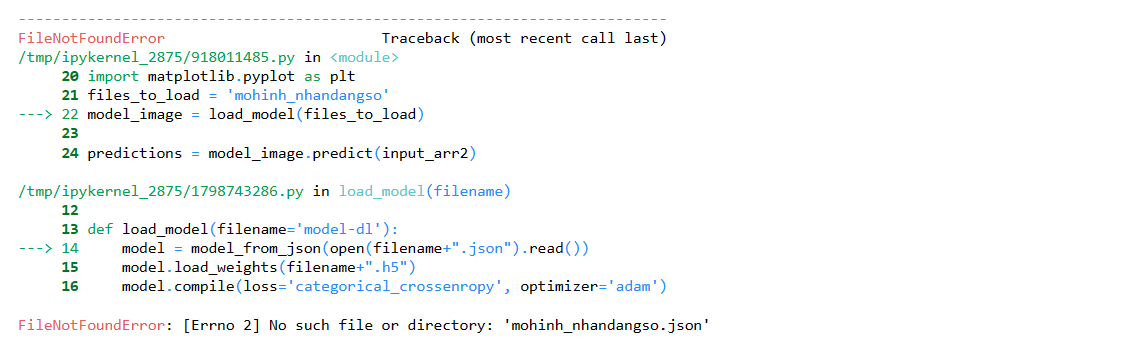
******

******

******

******

******

******

*Nhận xét: Sau khi xóa file json nhận thấy kết quả dự đoán bị lỗi do không có thư viện để load dữ liệu dự đoán*

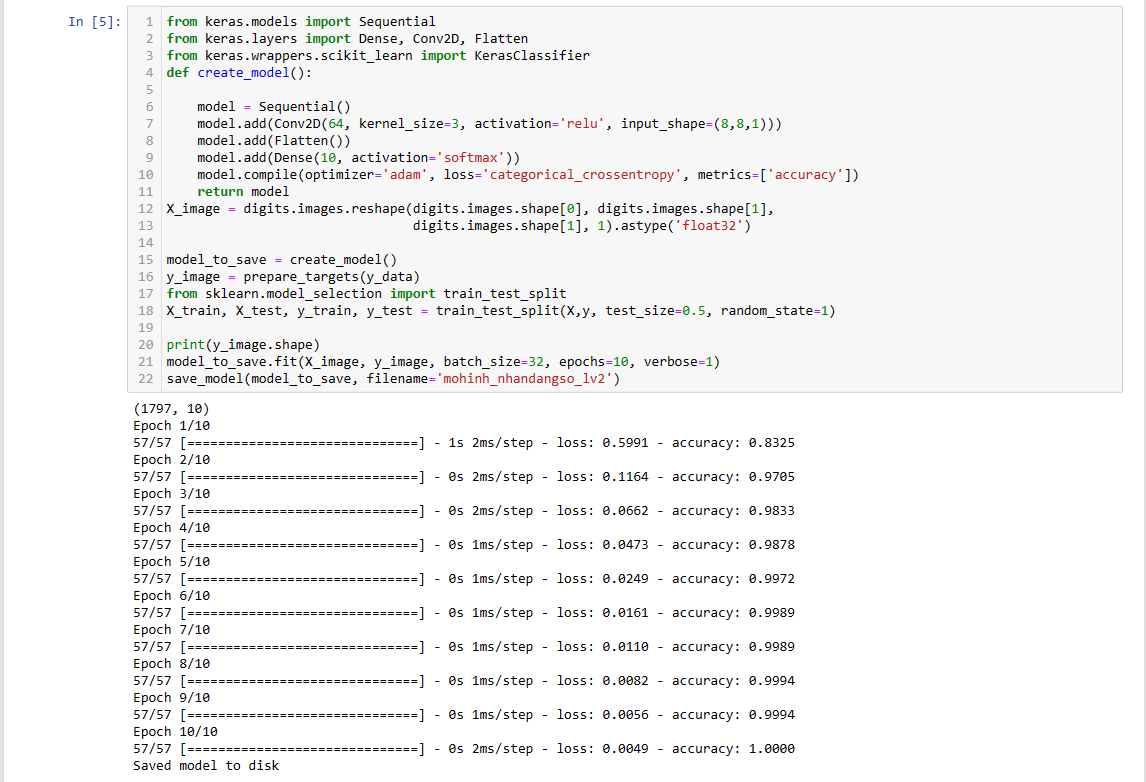
1. Ở phần “Lưu trữ mô hình học sâu”, ở chỗ load dữ liệu lên để huấn luyện, khối lệnh [6], bạn làm như chỗ khối lệnh [1] để thử tách dữ liệu làm 2 phần train/test với tỷ lệ 5/5 và bạn thử đưa tập train vào cho huấn luyện. So sánh kích thước các file mô hình được lưu trữ nếu chỉ học có ½ dữ liệu so với code ban đầu xem có sự khác nhau thế nào, và cho kết luận kích thước số mẫu để huấn luyện có ảnh hưởng đến dung lượng của mô hình được lưu không?

***Bài làm***

*Trả lời: Không*

******

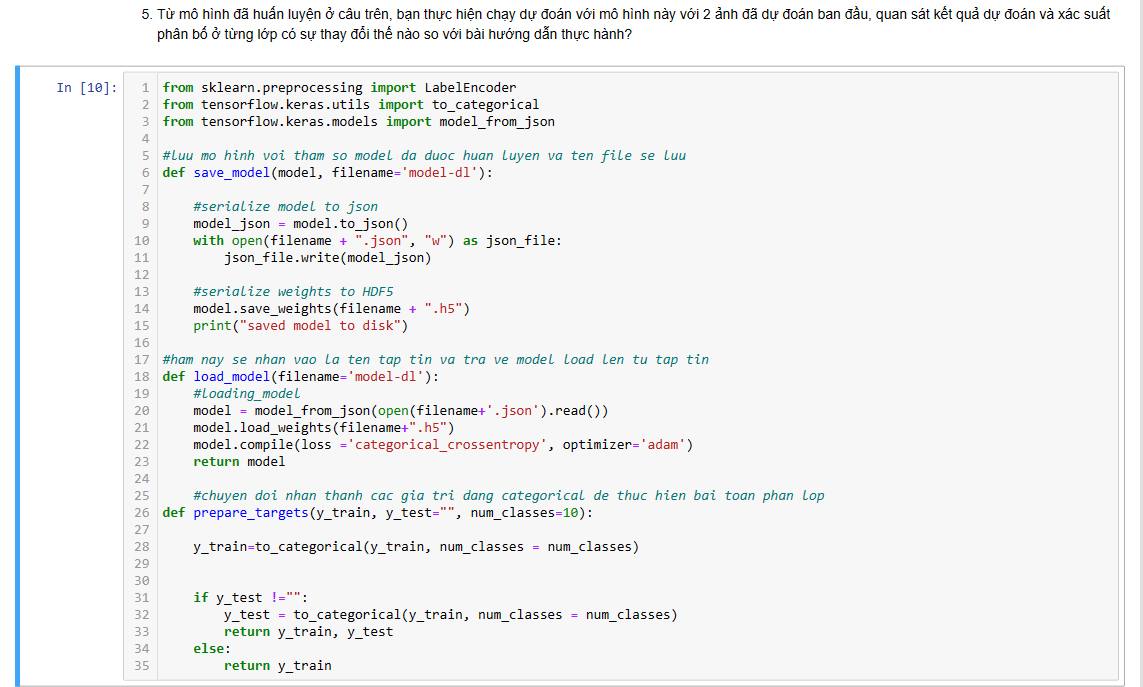
******

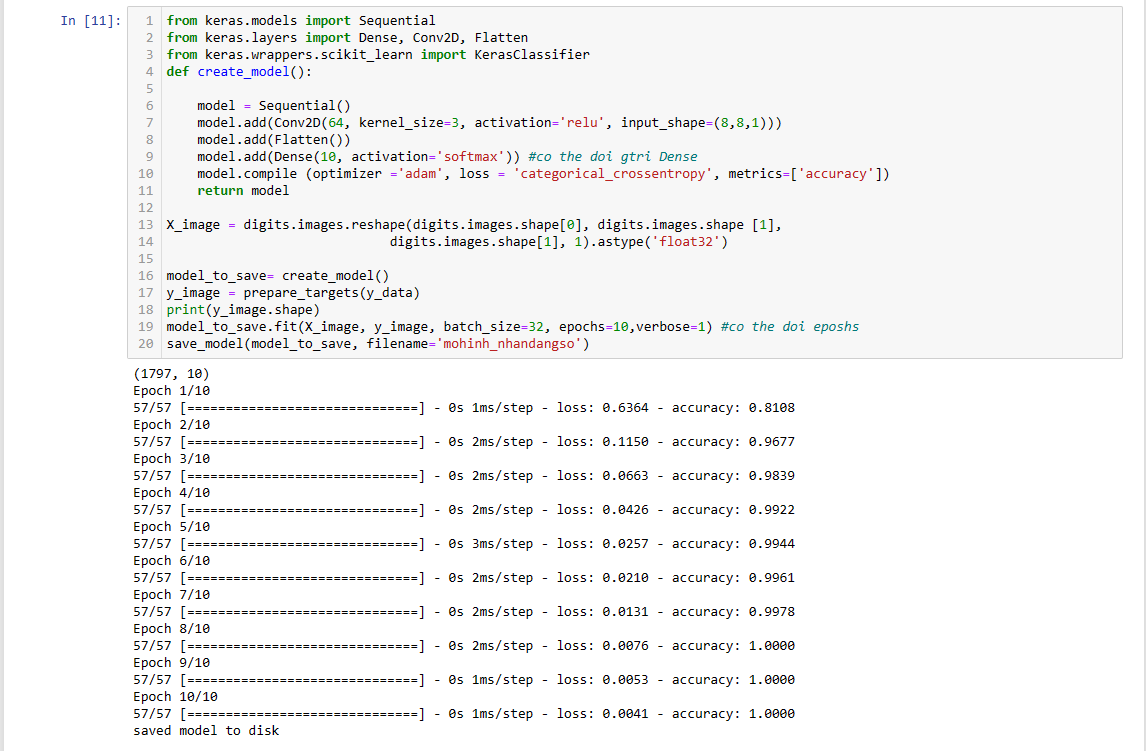
******

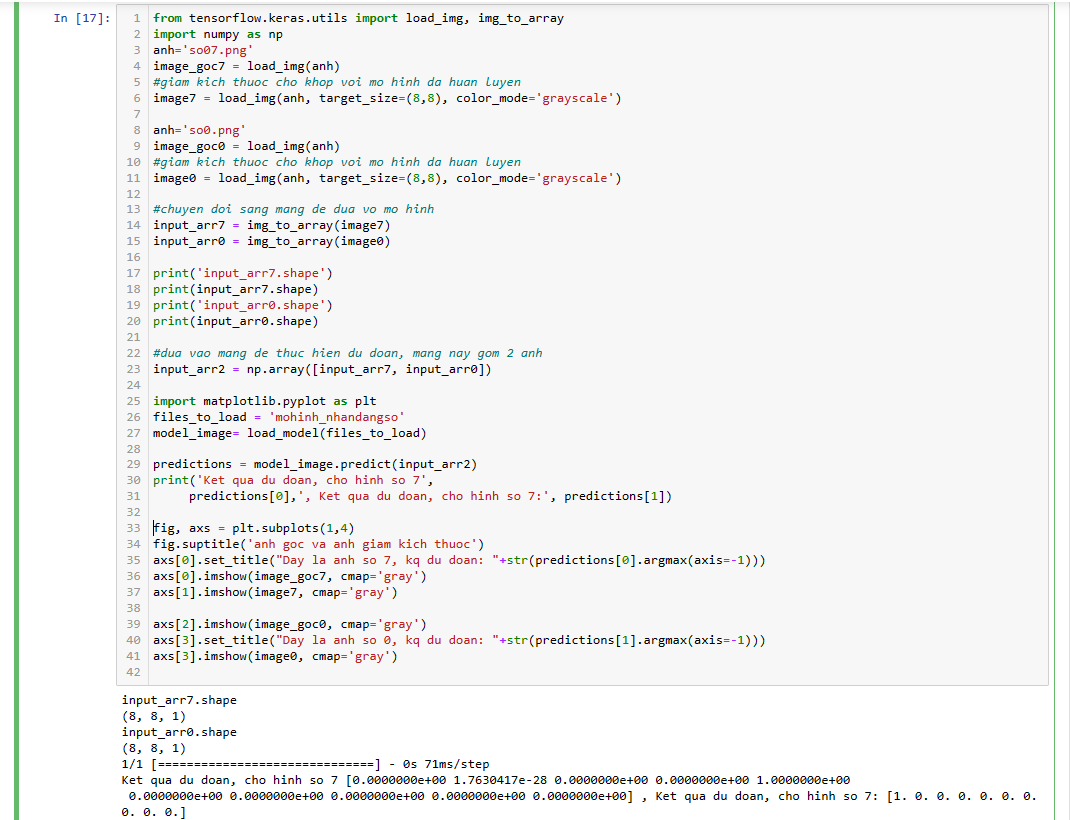
*Nhưng chỉ có độ chính xác và độ lỗi được cải thiện đáng kể*

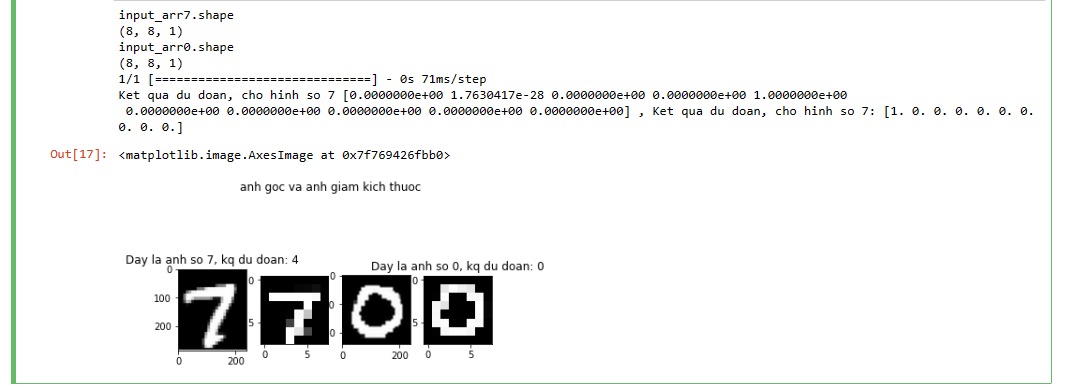
1. Từ mô hình đã huấn luyện ở câu trên, bạn thực hiện chạy dự đoán với mô hình này với 2 ảnh đã dự đoán ban đầu, quan sát kết quả dự đoán và xác suất phân bố ở từng lớp có sự thay đổi thế nào so với bài hướng dẫn thực hành?

***Bài làm***

******

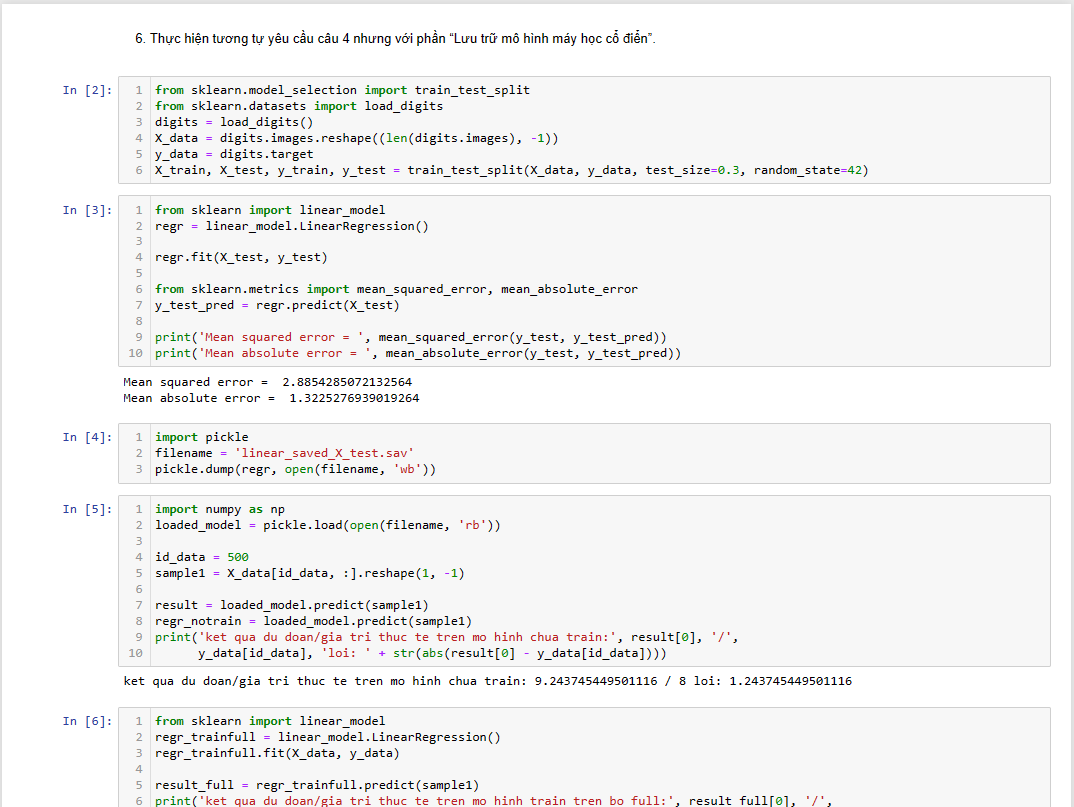
******

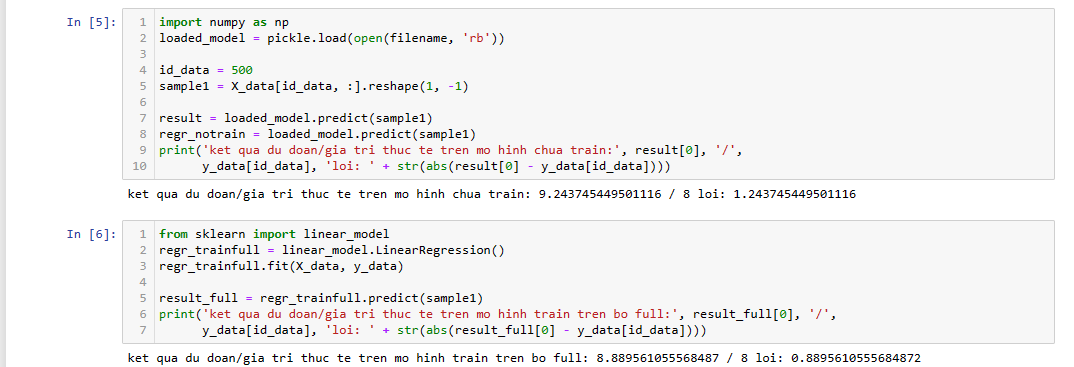
******

******

1. Thực hiện tương tự yêu cầu câu 4 nhưng với phần “Lưu trữ mô hình máy học cổ điển”.

***Bài làm***

******

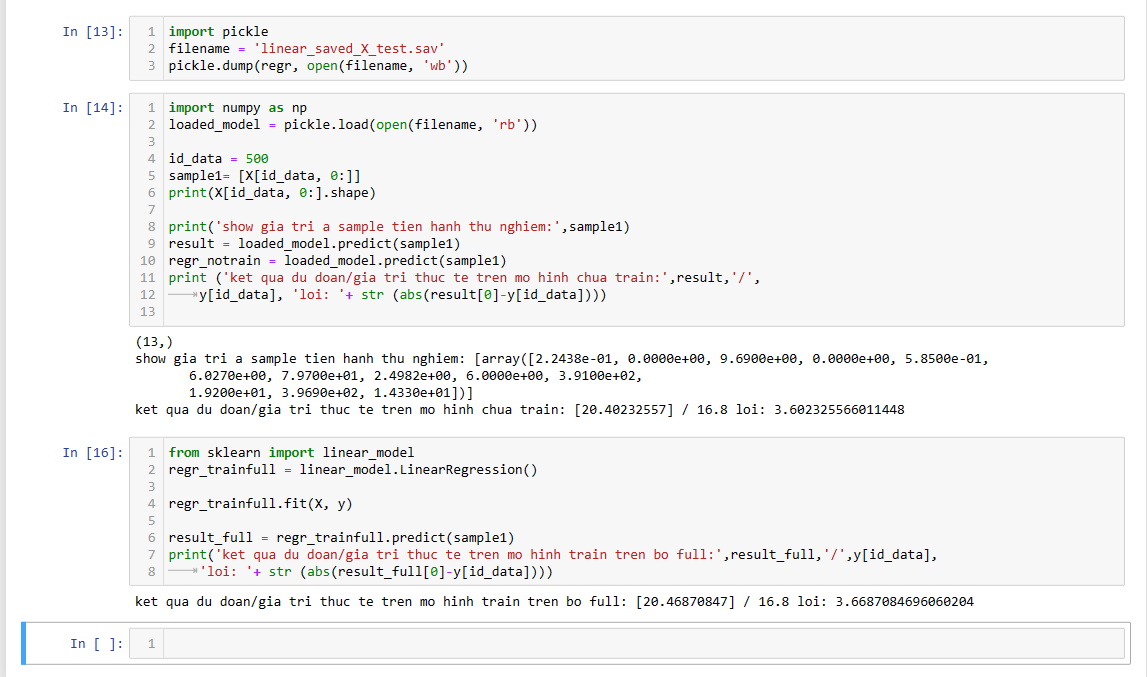
******

=> Kích thước số mẫu để huấn luyện có thể ảnh hưởng đến dung lượng của mô hình được lưu

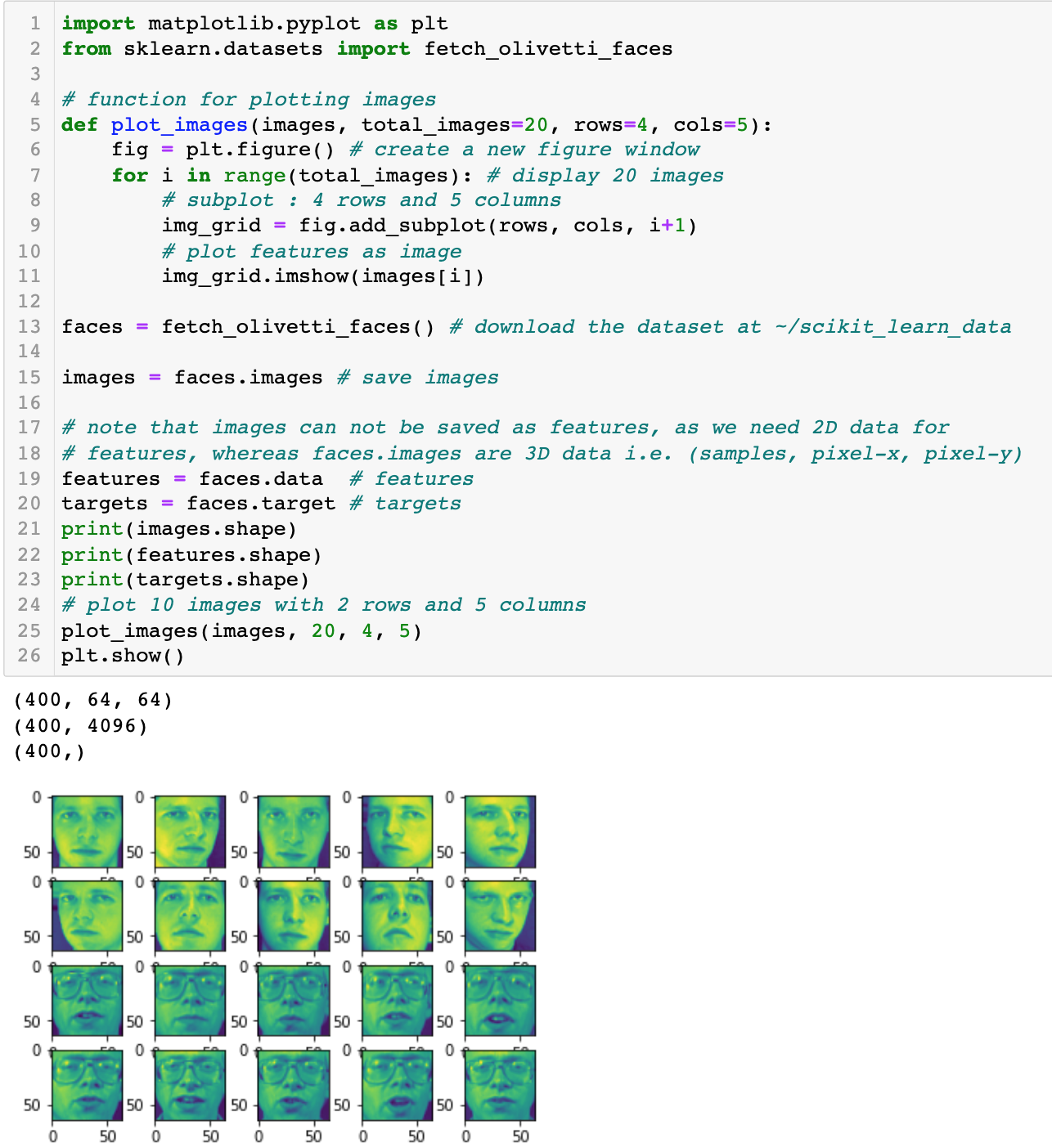
1. Dựa vào phần “Lưu trữ mô hình máy học cổ điển”, sử dụng dữ liệu customer.csv từ các bài trước, bạn hãy cho huấn luyện và lưu lại mô hình chạy trên tập customer.csv.

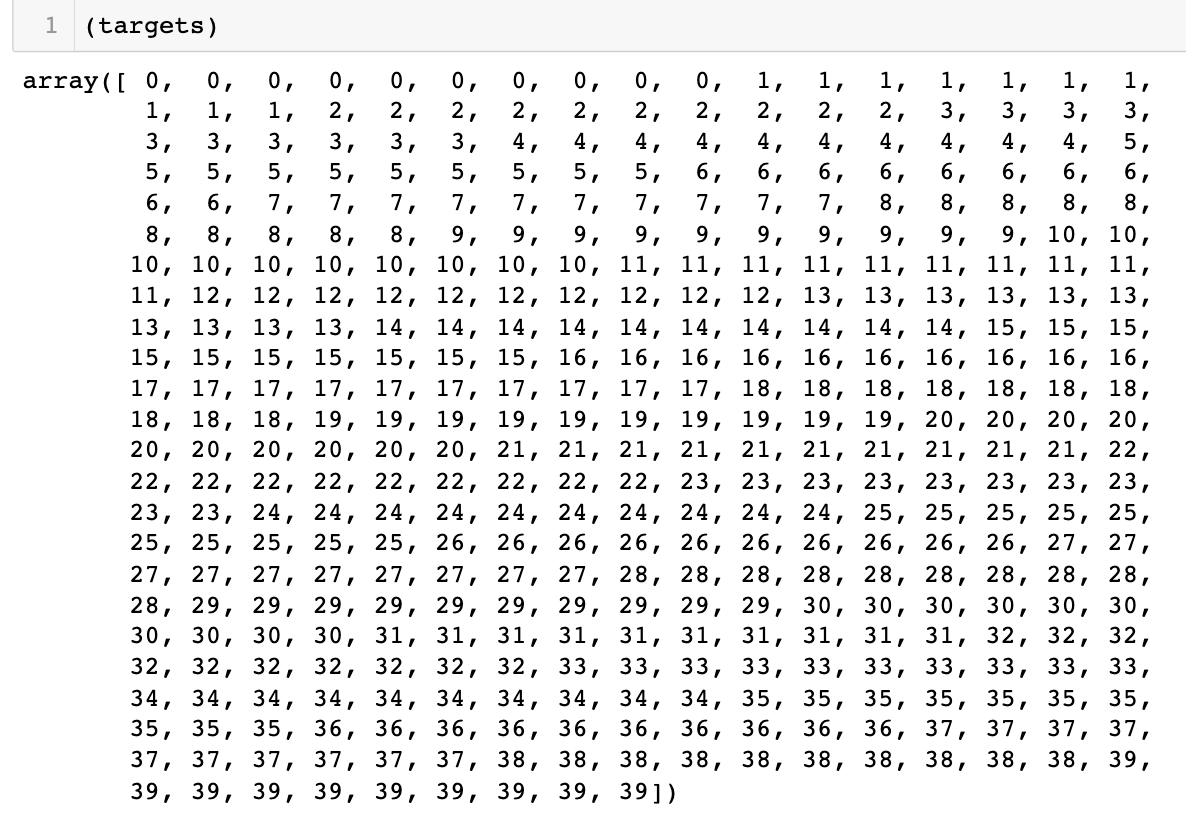
***Bài làm***

******

******

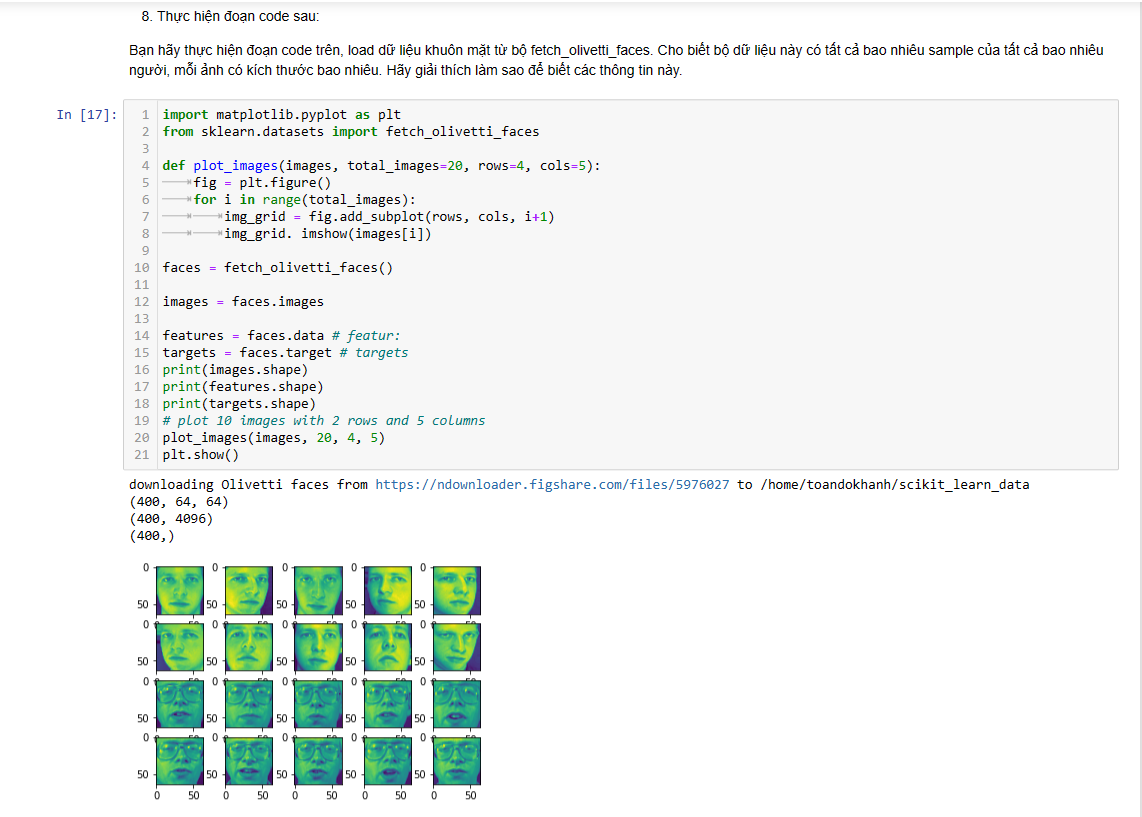
1. Thực hiện đoạn code[[4]](#footnote-4) sau:

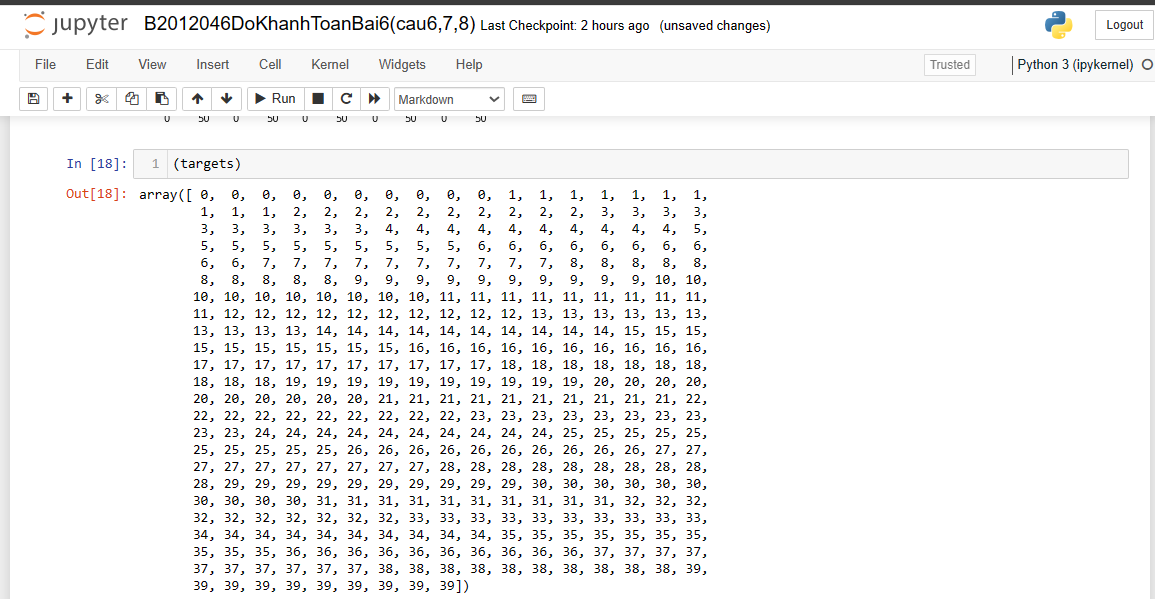




Bạn hãy thực hiện đoạn code trên, load dữ liệu khuôn mặt từ bộ **fetch\_olivetti\_faces**. Cho biết bộ dữ liệu này có tất cả bao nhiêu sample của tất cả bao nhiêu người, mỗi ảnh có kích thước bao nhiêu. Hãy giải thích làm sao để biết các thông tin này.

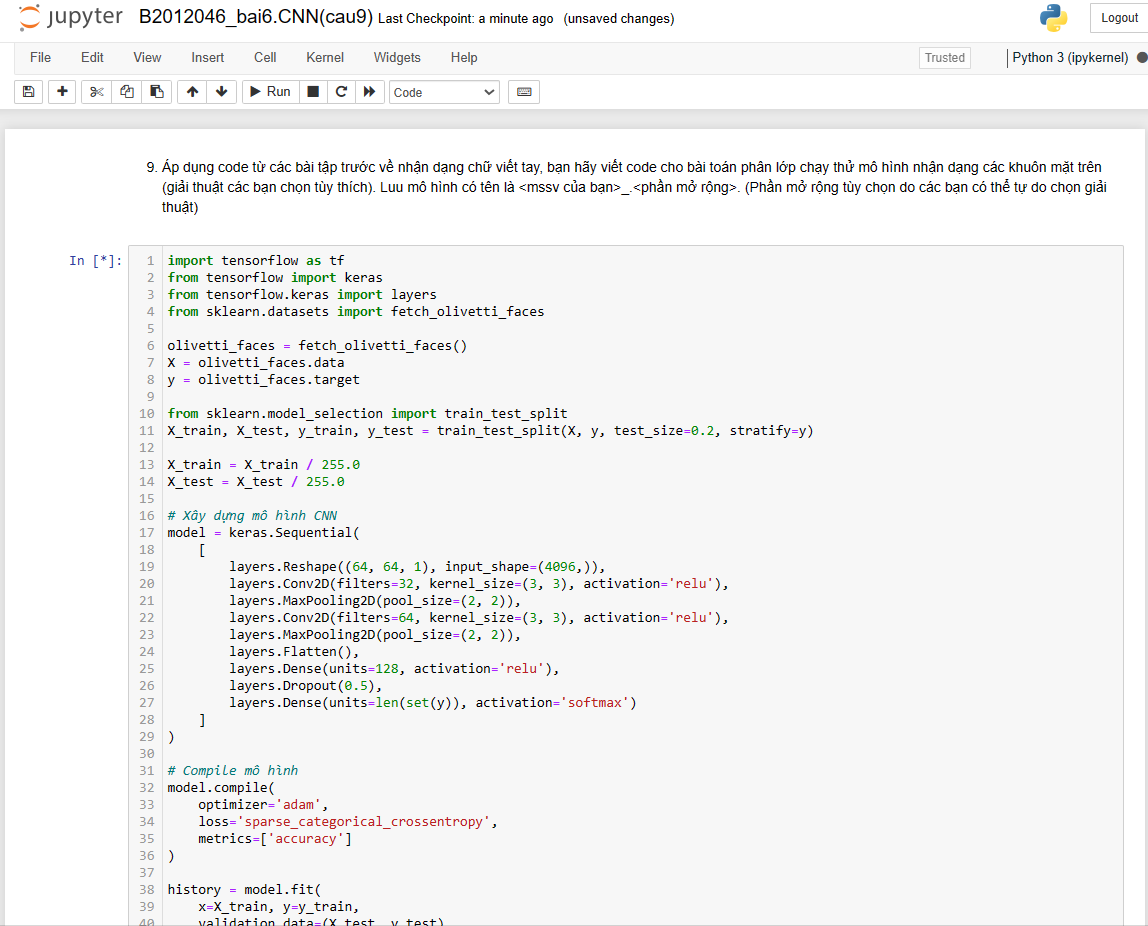
***Bài làm***

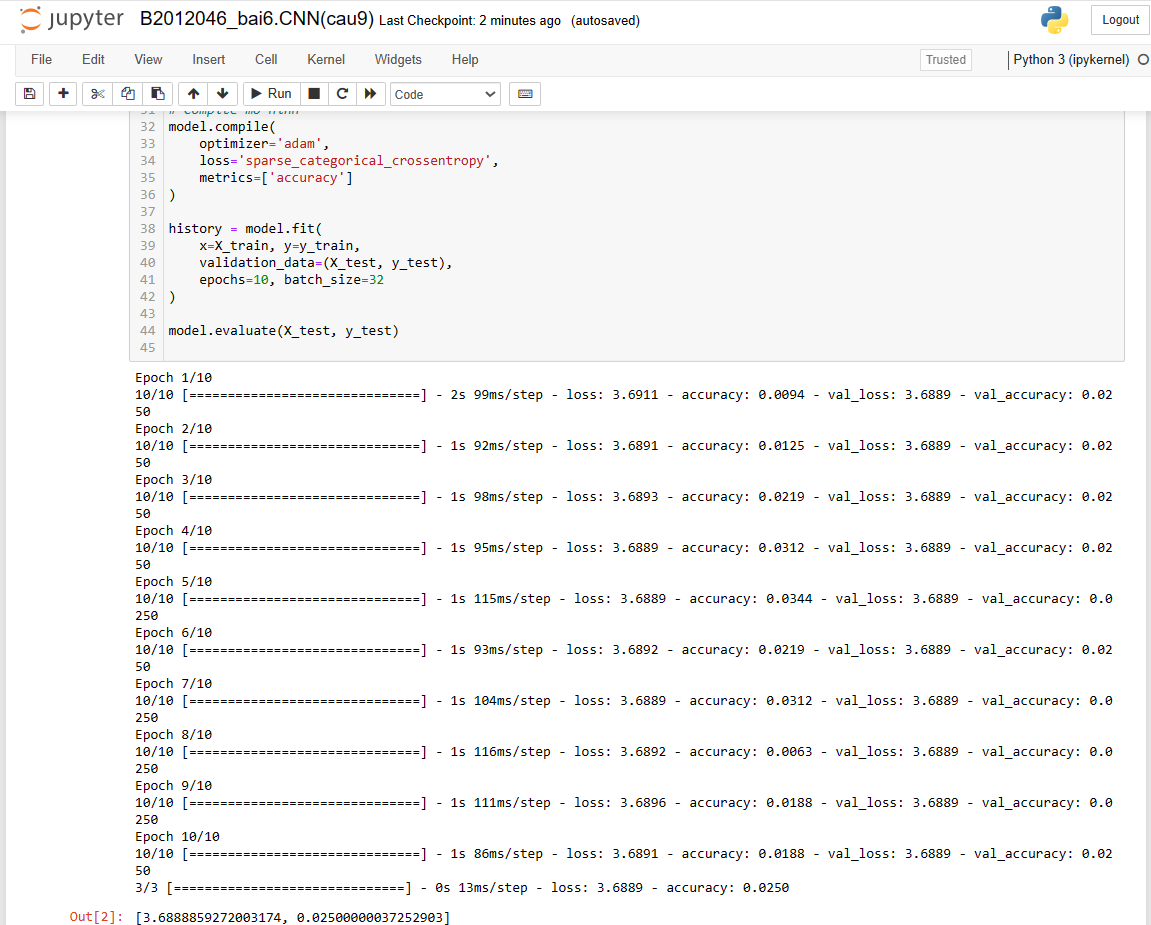
******

******

1. Áp dụng code từ các bài tập trước về nhận dạng chữ viết tay, bạn hãy viết code cho bài toán phân lớp chạy thử mô hình nhận dạng các khuôn mặt trên (giải thuật các bạn chọn tùy thích). Luu mô hình có tên là <mssv của bạn>\_<bai6>.<phần mở rộng>. (Phần mở rộng tùy chọn do các bạn có thể tự do chọn giải thuật)

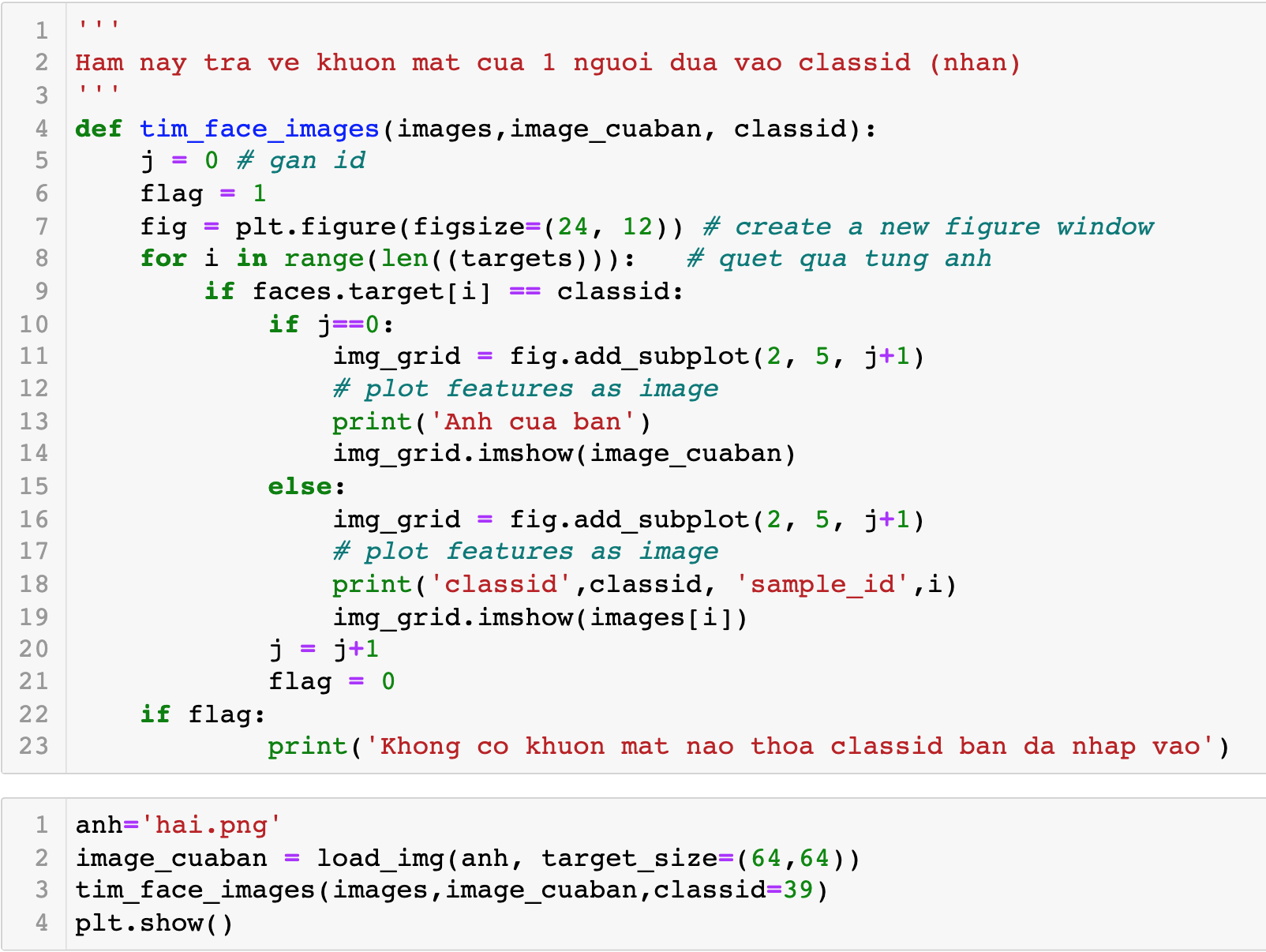
***Bài làm***

******

******

1. Bạn thử chụp hình khuôn mặt của bạn và chạy thử mô hình đã huấn luyện ở trên để xem thử gần giống với ai? Thể hiện kết quả.

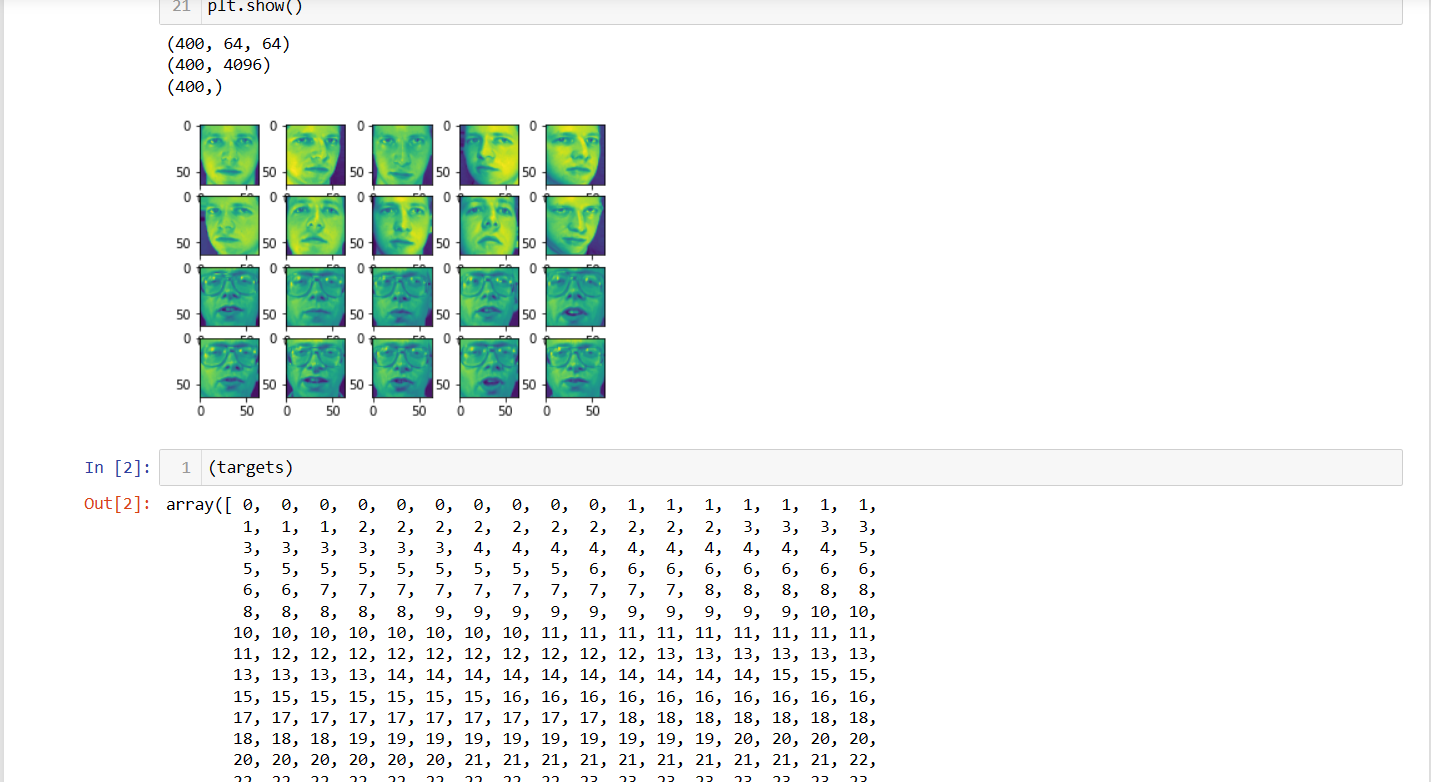
Sau khi biết được classid từ kết quả dự đoán. Bạn gõ khối lệnh như bên dưới để giúp bạn tìm được các khuôn mặt nếu biết được classid. Bố trí hình khuôn mặt của bạn kế bên và thử load khuôn mặt và bạn được nhận dạng vào classid để so sánh xem có sự giống nhau đến mức nào.



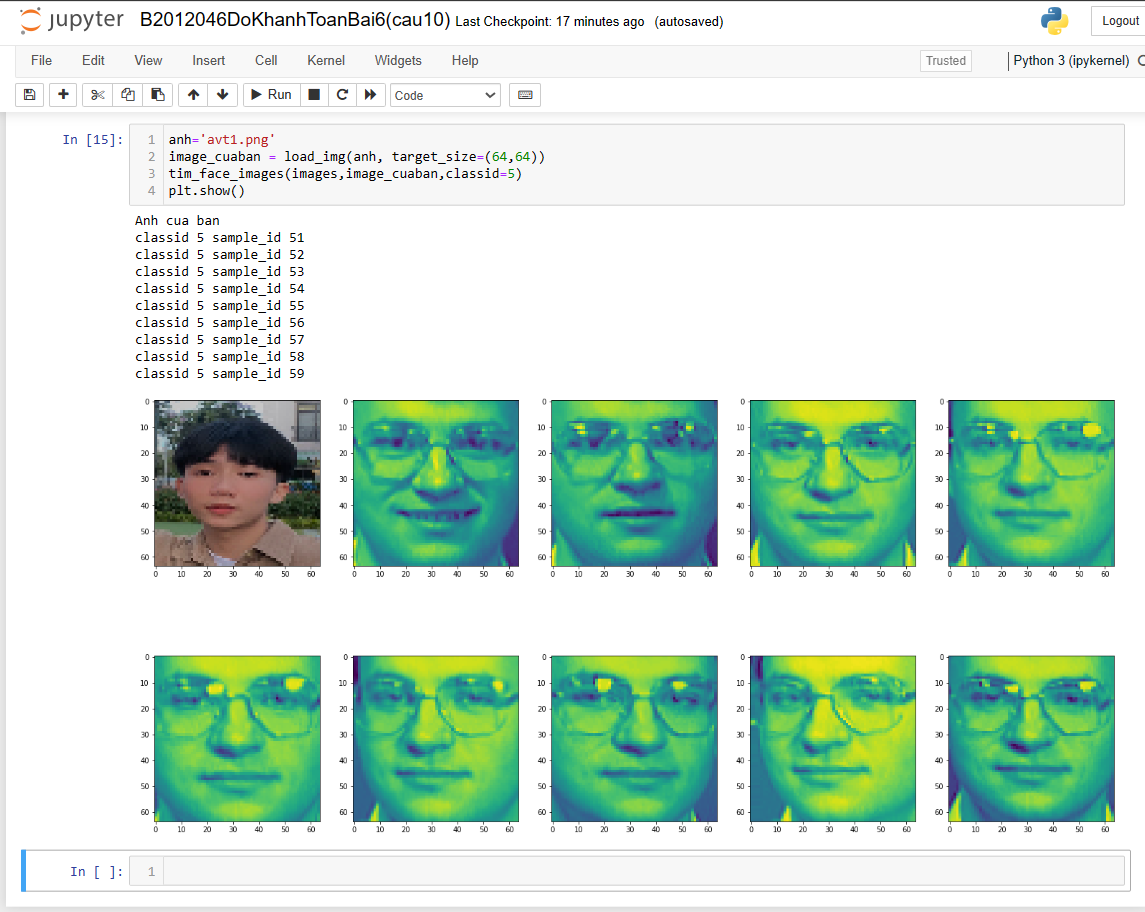


***Bài làm***

******

******

******

******

*Rất hy vọng bạn đã trải nghiệm giây phút thú vị khi coding!!*

*Cám ơn bạn đã tham gia môn học và chúc tất cả các bạn đều đạt kết quả tốt!*

**Chú ý:**

* Các bạn nộp file word: Quy tắc đặt tên file: **<mssv>-<hoten>-<bai><stt\_bai thực hành>.docx** nộp lên Classroom (VD: **B123456-NguyenVanA-bai1.docx**), kèm với các file khác được yêu cầu như phần câu hỏi đã nêu. **Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file zip**. File zip đặt tên như file word.
* Mỗi câu các bạn trả lời bằng hình hoặc dạng text tùy vào yêu cầu của câu hỏi và **TRẢ LỜI THEO ĐÚNG THỨ TỰ CÂU HỎI**. Nếu câu nào không trả lời được các bạn cứ để số thứ tự câu hỏi và bỏ trống phần trả lời.
* Các câu trả lời có tham khảo trên Internet phải trích dẫn link/nguồn.
* *Vi phạm 1 trong các điều sau đây bài thực hành sẽ bị 0 điểm:*
  + Đặt tên KHÔNG ĐÚNG quy tắc được yêu cầu.
  + Bài không đủ các thành phần (word, code+data (nếu có),...) đã được yêu cầu.
  + Bài không thực hiện đúng yêu cầu “**Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file .zip”**
  + Bị phát hiện copy, sao chép từ các bạn khác
  + Phần trả lời không ghi rõ trả lời cho câu nào
  + Thứ tự câu trả lời không đúng thứ tự câu hỏi

1. <https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/> [↑](#footnote-ref-1)
2. <https://www.kaggle.com/prmohanty/python-how-to-save-and-load-ml-models> [↑](#footnote-ref-2)
3. <https://machinelearningmastery.com/save-load-keras-deep-learning-models/> [↑](#footnote-ref-3)
4. https://mclguide.readthedocs.io/en/latest/sklearn/image.html [↑](#footnote-ref-4)