

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

&

**BÀI TẬP LỚN**

**Môn Học Máy**

**ĐỀ TÀI: Nhận diện chữ số viết bằng tay từ 0 đến 9.**

**Giảng viên hướng dẫn: Trần Anh Đạt**

**Nhóm thực hiện:**

1. **Trần Tuấn Minh - 63CNTT4**
2. **Hồ Công Thành - 63CNTT4**
3. **Trương Huy Phú - 63CNTT4**
4. **Nguyễn Việt Hoàng - 63CNTT4**

# 

**MỤC LỤC**

**1. GIỚI THIỆU VỀ HỌC MÁY........................................................................................................**

**2. CÁC PHƯƠNG PHÁP HỌC MÁY..............................................................................................**

**2.1. THUẬT TOÁN K-NEAREST NEIGHBORS (KNN) .……….............................................**

**2.2. THUẬT TOÁN LOGISTIC REGRESTION..………..........................................................**

**3. MÔ TẢ BÀI TOÁN NHẬN DIỆN CHỮ SỐ VIẾT BẰNG TAY TỪ 0 - 9.................................**

**3.1. XÂY DỰNG BÀI TOÁN BẰNG THUẬT TOÁN KNN.......................................................**

**3.2. XÂY DỰNG BÀI TOÁN BẰNG THUẬT TOÁN LOGISTIC REGRESTION ...............**

**4. KẾT LUẬN BÀI TOÁN VÀ HƯỚNG PHÁT TRIỂN................................................................**

**5. LỜI CẢM ƠN..................................................................................................................................**

## **1.** **Giới thiệu về học máy (machine learning)**

- Lịch sử và vai trò của machine learning:

Machine learning là thuật ngữ được đặt bởi Arthur Samuel vào năm 1959. Samuel là một IBMer người Mỹ kiêm nhà tiên phong trong lĩnh vực trí tuệ nhân tạo và máy tính chơi game. Năm 1960, thuật ngữ học máy phổ biến hơn thông qua cuốn sách của Nilsson, nội dung đề cập đến việc phân loại máy học.

Machine learning (học máy) có khởi nguồn từ lĩnh vực trí tuệ nhân tạo (AI). Đây là giải pháp có khả năng khiến các ứng dụng, phần mềm trở nên thông minh hơn mà không cần thực hiện bất kỳ mã hóa nào. Thông qua dữ liệu đầu vào, Machine learning có thể dự đoán các giá trị đầu ra mới.

- Ưu điểm và nhược điểm của học có giám sát.

Ưu điểm:

* Học có giám sát cho phép thu thập dữ liệu và tạo ra dữ liệu đầu ra từ những kinh nghiệm trước đó.
* Giúp tối ưu hóa các tiêu chí hiệu suất với sự trợ giúp của kinh nghiệm.
* ML có giám sát giúp giải quyết nhiều loại vấn đề tính toán trong thế giới thực.

Nhược điểm:

* Hạn chế của mô hình này là ranh giới quyết định có thể bị giới hạn nếu tập huấn luyện của bạn không có ví dụ mà bạn muốn có trong một lớp.

- Ưu điểm và nhược điểm của học không có giám sát.

Ưu điểm:

* Tất cả dữ liệu không được gắn nhãn và các thuật toán tìm hiểu cấu trúc vốn có từ dữ liệu đầu vào.

Nhược điểm:

* Phân loại dữ liệu lớn có thể là một thách thức. Đào tạo cho việc học có giám sát cần rất nhiều thời gian tính toán, vì vậy, nó đòi hỏi rất nhiều thời gian.

## **2. Phương pháp học máy**

**2.1 Thuật toán KNN (K-Nearest Neighbors).**

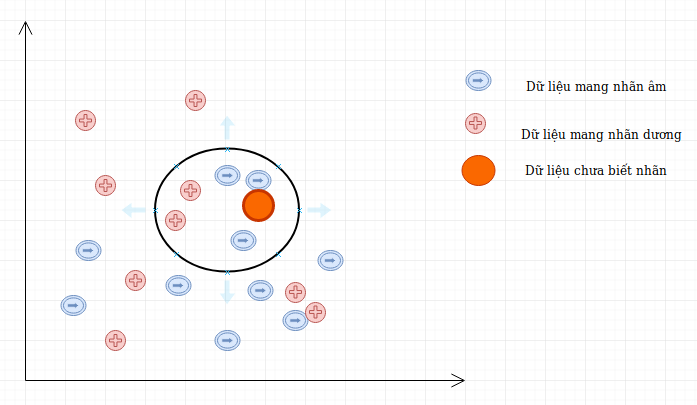
- **KNN** là một thuật toán học có giám sát, được sử dụng để phân loại hoặc hồi quy dữ liệu. Thuật toán này dựa trên ý tưởng rằng các dữ liệu tương tự nhau sẽ nằm gần nhau trong không gian đặc trưng.  Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

**- Ý tưởng của KNN**

Thuật toán KNN hoạt động bằng cách tìm **k** điểm gần nhất với điểm dữ liệu cần dự đoán. Sau đó, điểm dữ liệu đó sẽ được gán nhãn của đa số các điểm kề cận.

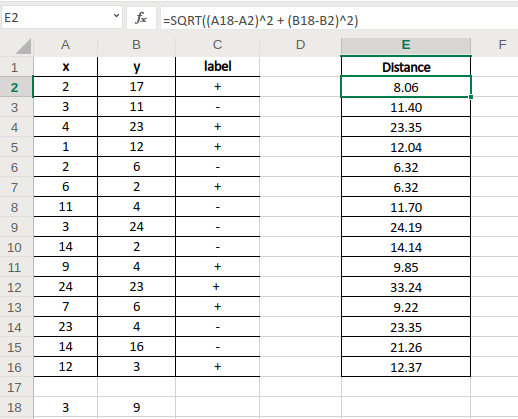
**Ví dụ:**

Chúng ta có một tập dữ liệu (D = 15 dữ liệu ) về các điểm dữ liệu đã được phân loại thành 2 nhãn (+) và (-) và mô tả bởi các đặc trưng như màu sắc, kích thước, hình dạng.... được biểu diễn trên trục tọa độ như hình vẽ.Chúng ta có thể sử dụng thuật toán KNN để phân loại điểm dữ liệu A mới có 2 đặc trưng (3;9 ) mà chúng ta không biết tên.



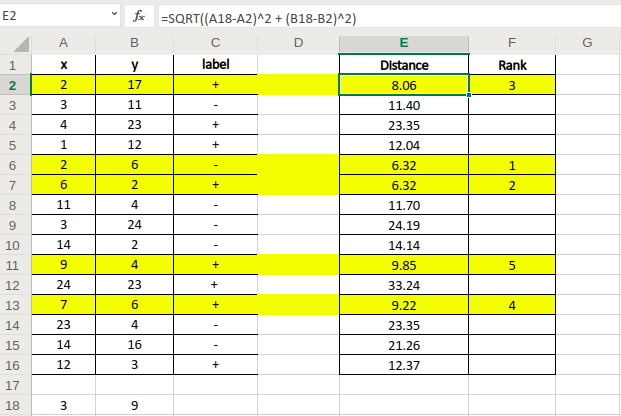
Chúng ta sẽ tính khoảng cách từ A đến các điểm dữ liệu.

Distance = sqrt[ (x[A] – x[i])^2 + (y[A] – y[i])^2 ]



Sau khi tính xong,chúng ta sẽ tìm K điểm gần nhất với điểm mới A (K là tham số mà chúng ta định nghĩa) khoảng cách nhỏ nhất.Giả sự K=5.

Tiếp theo,chúng ta sẽ đếm xem trong 5 điểm đó có bao nhiêu điểm thuộc nhãn (+) và bao nhiêu điểm thuộc nhãn (-).



Trong 5 điểm ta thấy có 4 điểm mang nhãn (+) và 1 điểm mang nhãn (-).

Vậy ta có thể đưa ra kết luận là điểm A cần dự đoán mang nhãn (+).

**Cách thức hoạt động của KNN**

Thuật toán KNN có thể được chia thành các bước sau:

1. Chuẩn bị dữ liệu: Dữ liệu được chia thành hai tập: tập huấn luyện và tập kiểm tra. Tập huấn luyện được sử dụng để đào tạo thuật toán, trong khi tập kiểm tra được sử dụng để đánh giá hiệu suất của thuật toán.
2. Chọn giá trị k: Giá trị k là số lượng điểm kề cận cần được tìm. Giá trị k thường được đặt là một số nhỏ, chẳng hạn như 3 hoặc 5.
3. Tìm k điểm gần nhất: Sử dụng một số phương thức để tìm k điểm gần nhất với điểm dữ liệu cần dự đoán. Các phương thức phổ biến bao gồm khoảng cách Euclid, khoảng cách Manhattan và khoảng cách Minkowski.
4. Phân loại hoặc hồi quy: Dựa trên nhãn của các điểm kề cận, chúng ta có thể phân loại hoặc hồi quy điểm dữ liệu cần dự đoán.

**Ưu điểm của KNN**

* KNN là một thuật toán đơn giản và dễ hiểu.
* KNN có thể được sử dụng cho cả phân loại và hồi quy.
* KNN có thể hoạt động tốt với dữ liệu không gian rời rạc.

**Nhược điểm của KNN**

* KNN có thể không hiệu quả với dữ liệu lớn.
* KNN có thể nhạy cảm với giá trị k.

**Ứng dụng của KNN**

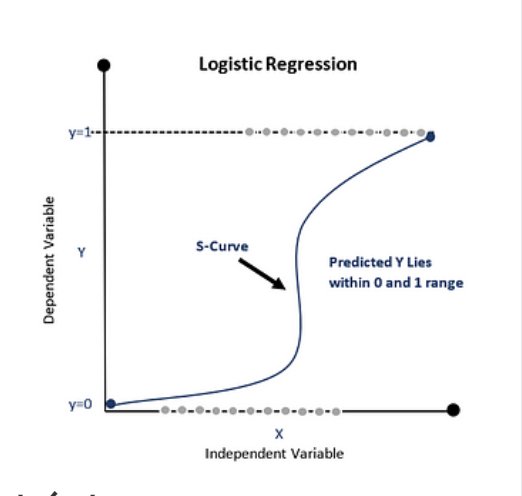
Thuật toán KNN có thể được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm:

* Phân loại hình ảnh
* Nhận dạng giọng nói
* Phân tích thị trường
* Phân loại khách hàng
* Phân tích dữ liệu y tế

**Kết luận**

Thuật toán KNN là một thuật toán học máy đơn giản và hiệu quả, có thể được sử dụng cho nhiều ứng dụng khác nhau.

**2.2 Thuật Toán hồi quy logistic**

Hồi quy logistic là một thuật toán học máy có giám sát,được sử dụng để phân loại dữ liệu nhị phân. Trong phân loại nhị phân, có hai kết quả có thể xảy ra, chẳng hạn như "có" hoặc "không", "chính xác" hoặc "sai", hoặc "lừa đảo" hoặc "không lừa đảo". Hồi quy logistic sử dụng hàm logistic để ước tính xác suất của một đối tượng thuộc một lớp nhất định. 

Mô hình hồi quy logistic được biểu diễn dưới dạng hàm tuyến tính, với một hàm sigmoid được áp dụng cho đầu ra để biến nó thành một giá trị trong khoảng từ 0 đến 1. Giá trị này được giải thích là xác suất của đối tượng thuộc lớp dương.

Công thức toán học của hồi quy logistic như sau:

P(Y = 1 | X) = 1 / (1 + e^(-w^T X))

Trong đó:

* P(Y = 1 | X) là xác suất của đối tượng thuộc lớp dương, cho một tập dữ liệu đầu vào X.
* w là vector trọng số của mô hình.
* X là tập dữ liệu đầu vào.
* e là số e.

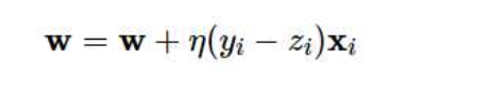
Để đào tạo mô hình hồi quy logistic, chúng ta cần tối ưu hóa hàm mất mát, thường là hàm mất mát cross-entropy.

L(w) = -\sum\_{i=1}^n log(P(Y\_i = 1 | X\_i))

Trong đó:

* L(w) là hàm mất mát của mô hình.
* n là số mẫu trong tập dữ liệu.
* Y\_i là nhãn của mẫu thứ i.
* X\_i là tập dữ liệu đầu vào của mẫu thứ i.

Hàm cập nhật:



Có nhiều phương pháp khác nhau để tối ưu hóa hàm mất mát, chẳng hạn như gradient descent, stochastic gradient descent, và LBFGS.

Hồi quy logistic là một thuật toán học máy phổ biến được sử dụng trong nhiều ứng dụng, chẳng hạn như:

* Phân loại email spam.
* Phân loại hình ảnh.
* Phân loại văn bản.
* Phát hiện gian lận.
* Dự đoán điểm số sinh viên.

Ưu điểm của hồi quy logistic:

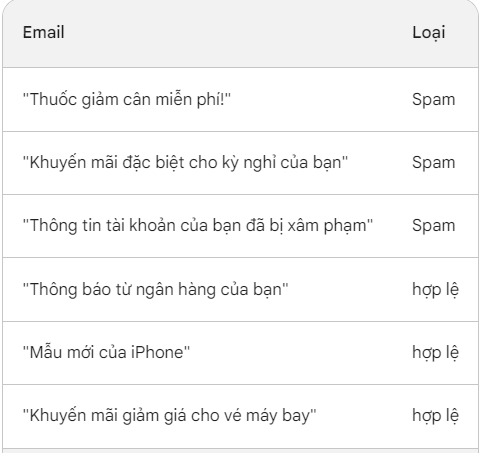
* Đơn giản để hiểu và triển khai.
* Có thể giải thích được.
* Có thể được áp dụng cho các tập dữ liệu lớn.

Nhược điểm của hồi quy logistic:

* Có thể không chính xác cho các tập dữ liệu không cân bằng.
* Có thể dễ bị overfitting.

Dưới đây là một số ví dụ về cách sử dụng hồi quy logistic trong thực tế:

* Phân loại email spam: Một mô hình hồi quy logistic có thể được đào tạo trên một tập dữ liệu email, bao gồm cả email spam và email hợp lệ. Mô hình sẽ học cách phân biệt giữa hai loại email dựa trên các thuộc tính của email, chẳng hạn như chủ đề, nội dung và địa chỉ người gửi.



* Một công ty bảo hiểm có thể sử dụng hồi quy logistic để phân loại khách hàng có khả năng gặp tai nạn hay không.
* Một công ty bán lẻ có thể sử dụng hồi quy logistic để phân loại khách hàng có khả năng mua sản phẩm hay không.

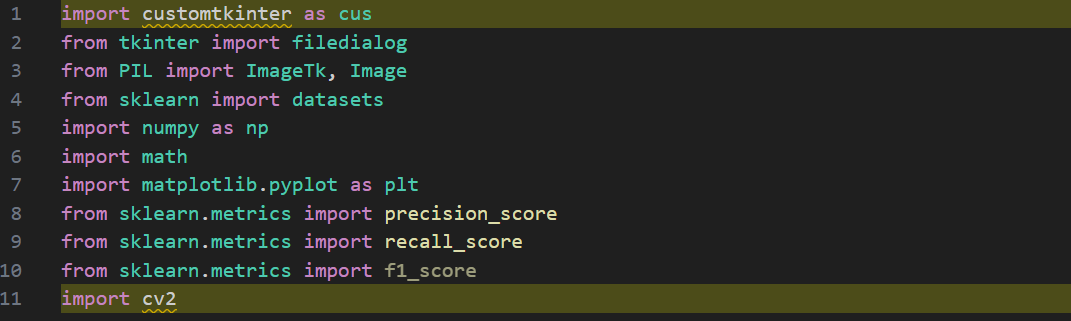
Thuật toán hồi quy logistic là một trong những thuật toán được sử dụng rộng rãi có thể được triển khai để thực hiện các dự đoán khác nhau. Tuy nhiên, chúng ta có xu hướng thu được một kết quả rời rạc từ thuật toán hồi quy logistic. Ngoài ra, thuật toán hồi quy logistic yêu cầu sức mạnh tính toán thấp do tính đơn giản của nó. Do đó, thuật toán này có thể được coi là một mô hình chuẩn để đo lường hiệu suất.

**3. Mô tả bài toán**

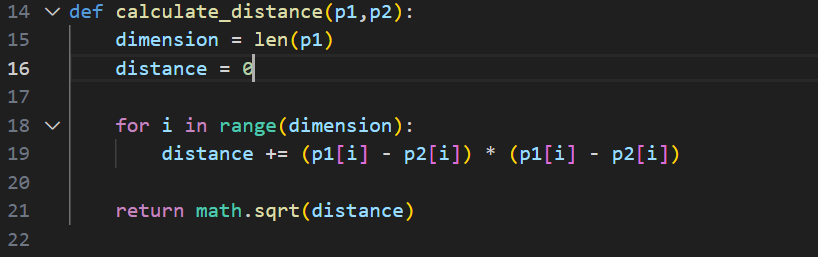
Bài toán: Nhận diện chữ số viết tay từ 0 – 9.

* Phạm vi dữ liệu: Bộ dữ liệu bao gồm 1797 hình ảnh chữ số viết tay có dạng nền đen chữ trắng.
* Các thuật toán áp dụng:
  + K-Nearest Neighbors
  + Logistics Regrestion
* Đây là bài toán phân tích theo cụm, do đó hai thuật toán trên phù hợp với bài toán đưa ra.
* Công cụ áp dụng: ngôn ngữ lập trình Python.

**3.1 - Xây dựng bằng thuật toán KNN**

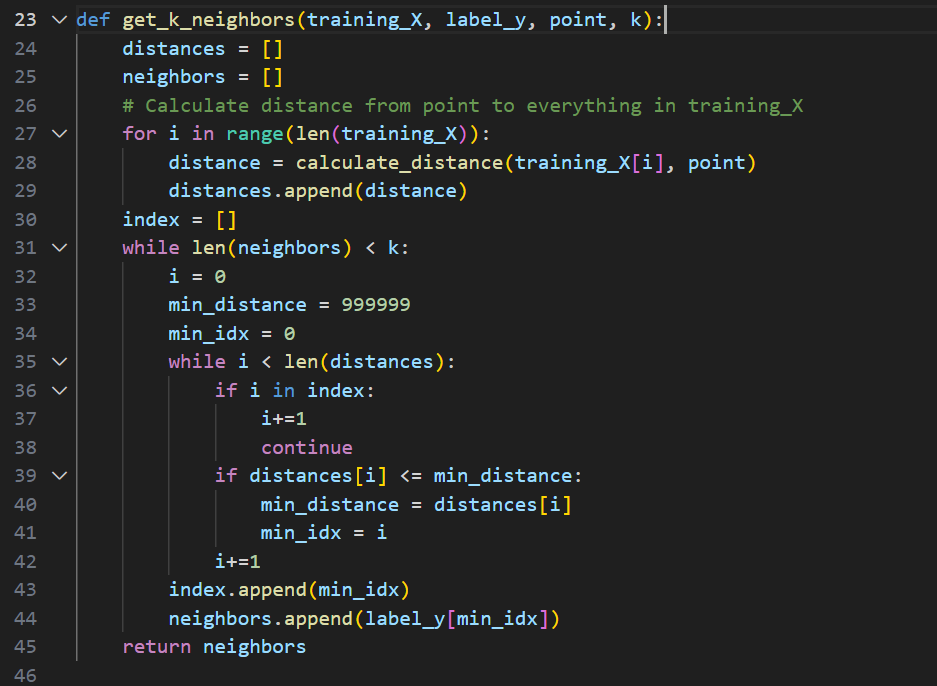


Các dòng code từ 1 đến 11: Import các thư viện cần thiết, bao gồm **customtkinter, tkinter, PIL, NumPy, scikit-learn và OpenCV.**

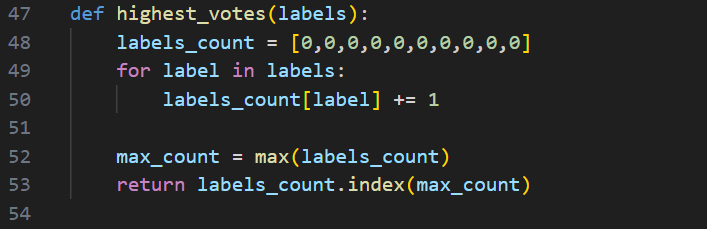


Các dòng code từ 14 đến 22: Định nghĩa **hàm calculate\_distance(),** tính toán khoảng cách giữa hai điểm trong không gian đặc trưng nhất định.Trong bài toán chúng em là không gian 64 chiều tương ứng với 64 đặc trưng.Và khoảng cách từ điểm dữ liệu mới (A) tới các điểm dự liệu trong tập dữ liệu (D) sẽ được tính theo công thức:

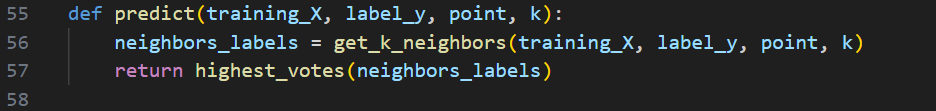
**Distance = Sqrt{(A[1] – D[i1])^2 + [(A[2] – D[i2])^2 +……+[(A[64] – D[i64])^2}**



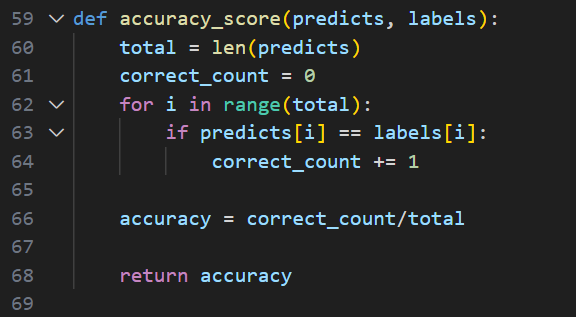
Các dòng code từ 23 đến 46: Định nghĩa hàm **get\_k\_neighbors(),** nhận đầu vào là tập huấn luyện dữ liệu, tập nhãn, điểm và giá trị **k** và trả về danh sách các điểm huấn luyện gần nhất với điểm cho trước.



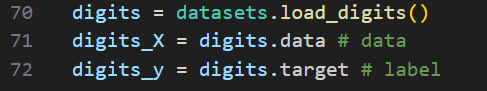
Các dòng code từ 47 đến 54: Định nghĩa hàm **highest\_votes()**, nhận đầu vào là danh sách nhãn và trả về nhãn có tần suất xuất hiện cao nhất.Trong **k** điểm có **Distance** nhỏ nhất vừa lọc ra,sẽ đếm xem nhãn dán nào nhiều nhất thì **return** ra nhãn dán đó.

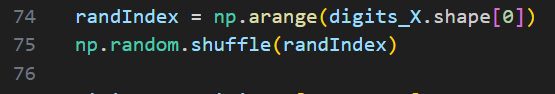


Các dòng code từ 55 đến 58: Định nghĩa hàm **predict()**, nhận đầu vào là tập huấn luyện dữ liệu, tập nhãn, điểm và giá trị k và trả về nhãn dự đoán cho điểm cho trước bằng thuật toán **K-Means**.



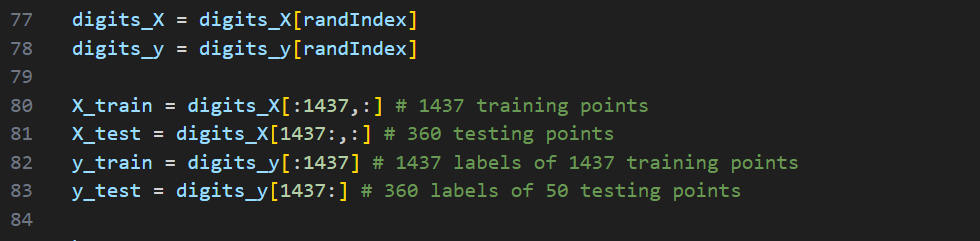
Các dòng code từ 59 đến 69: Định nghĩa hàm **accuracy\_score()**, nhận đầu vào là danh sách nhãn dự đoán và danh sách nhãn thật và trả về độ chính xác của các dự đoán.



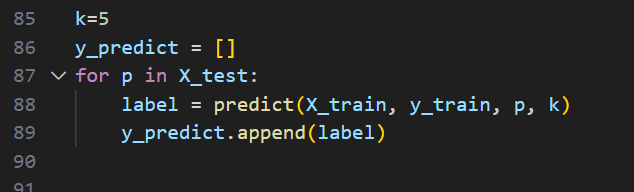


Tải tập dữ liệu Digits từ scikit-learn.

Sau đó xáo trộn dữ liệu và chia thành tập huấn luyện và tập kiểm tra.

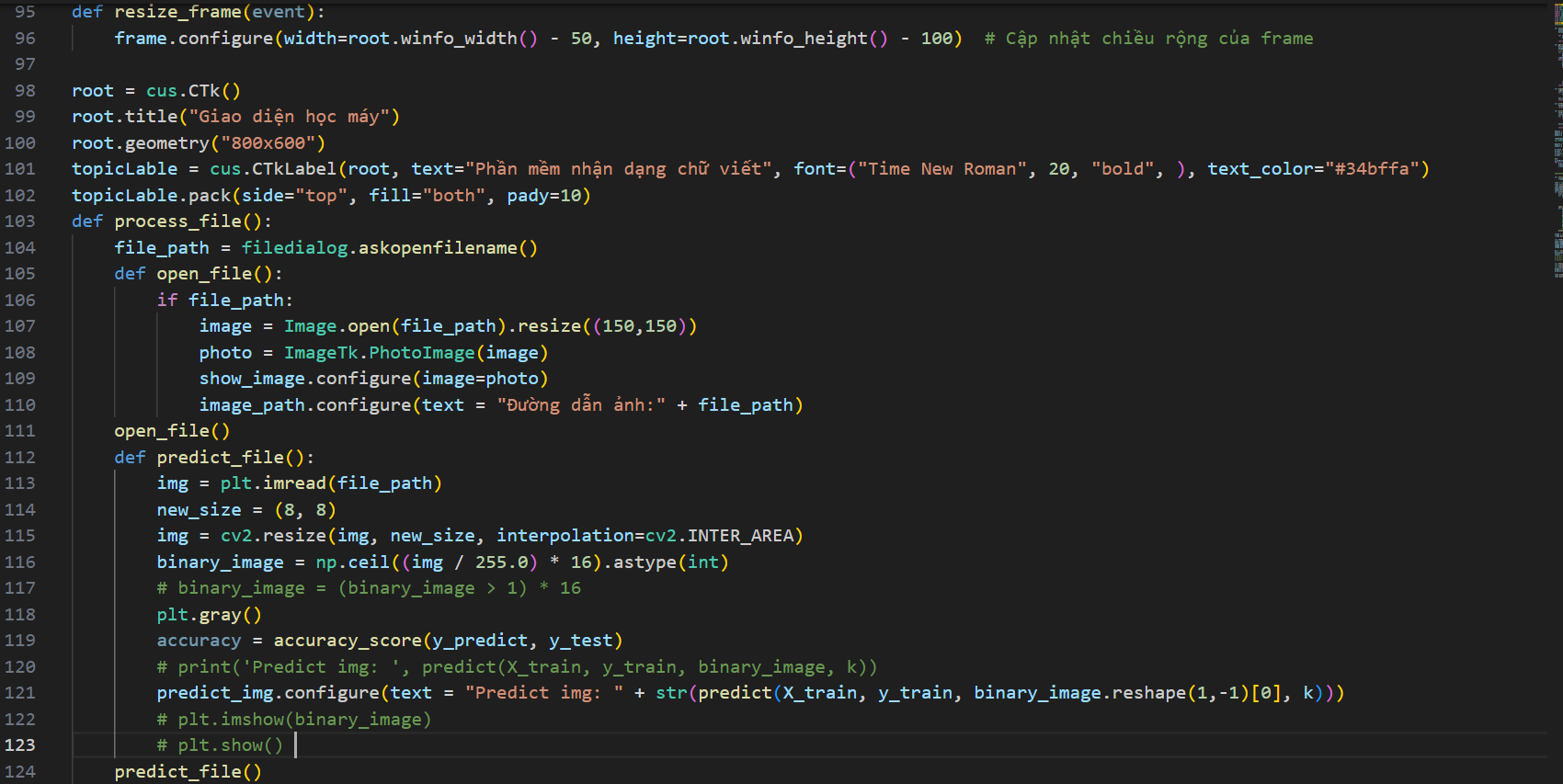


Các dòng code từ 77 đến 84: Đào tạo thuật toán KNN trên tập huấn luyện.



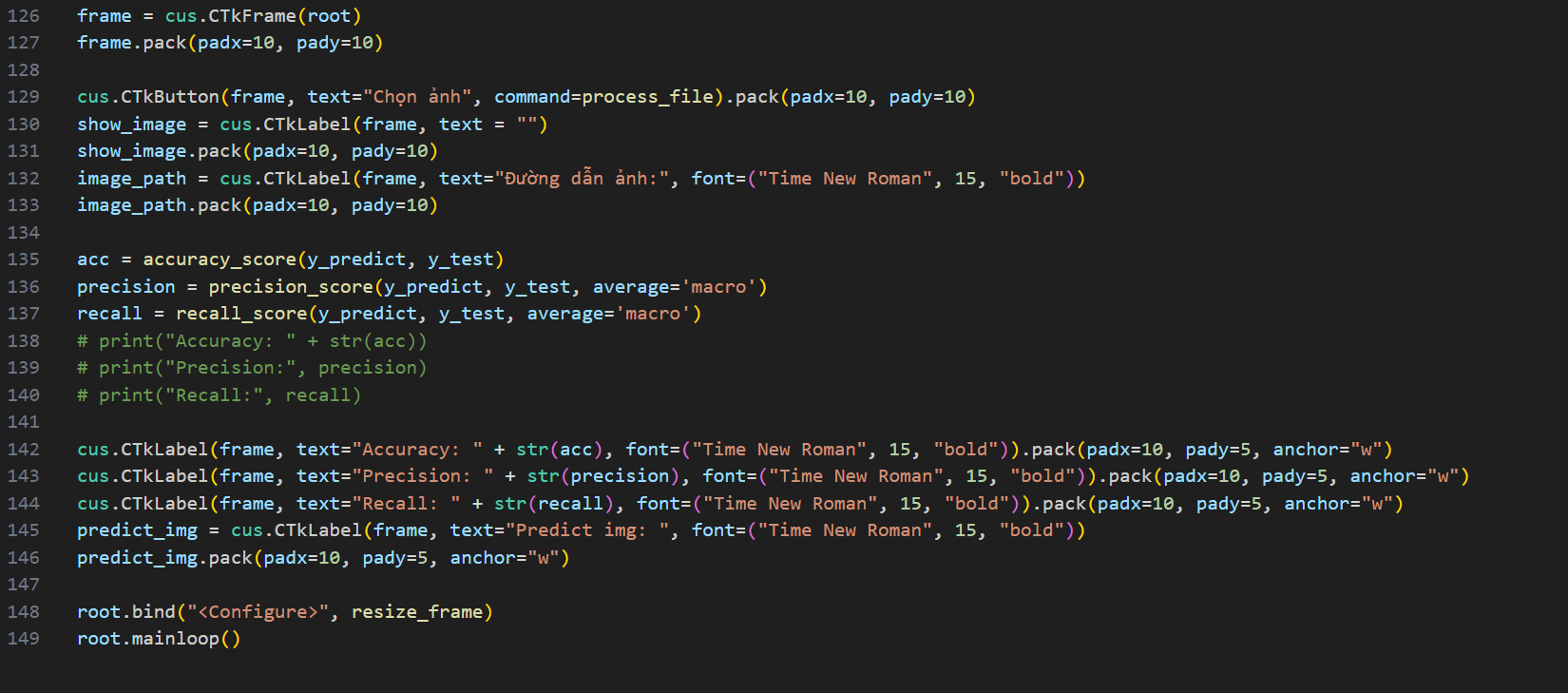
Cho k = 5 :

Và dự đoán nhãn trên tập kiểm tra và tính toán độ chính xác của các dự đoán.



Tạo giao diện người dùng (GUI) bằng thư viện customtkinter.

Thêm các nút và nhãn vào GUI để người dùng có thể chọn tệp hình ảnh và xem kết quả dự đoán.

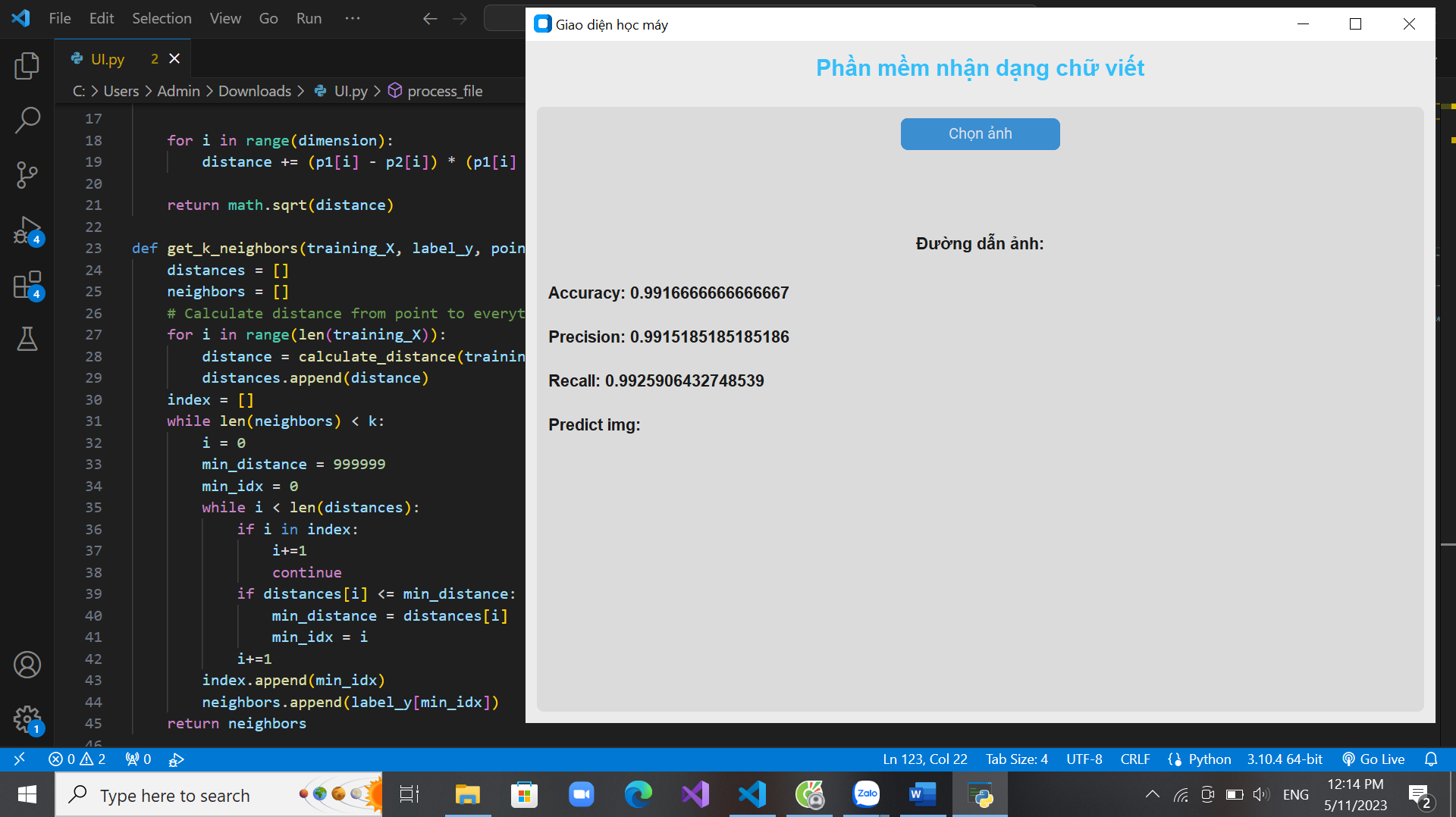


Liên kết hàm resize\_frame() với sự kiện <Configure> của cửa sổ gốc. Hàm này đảm bảo rằng khung luôn được thay đổi kích thước để vừa với cửa sổ gốc.

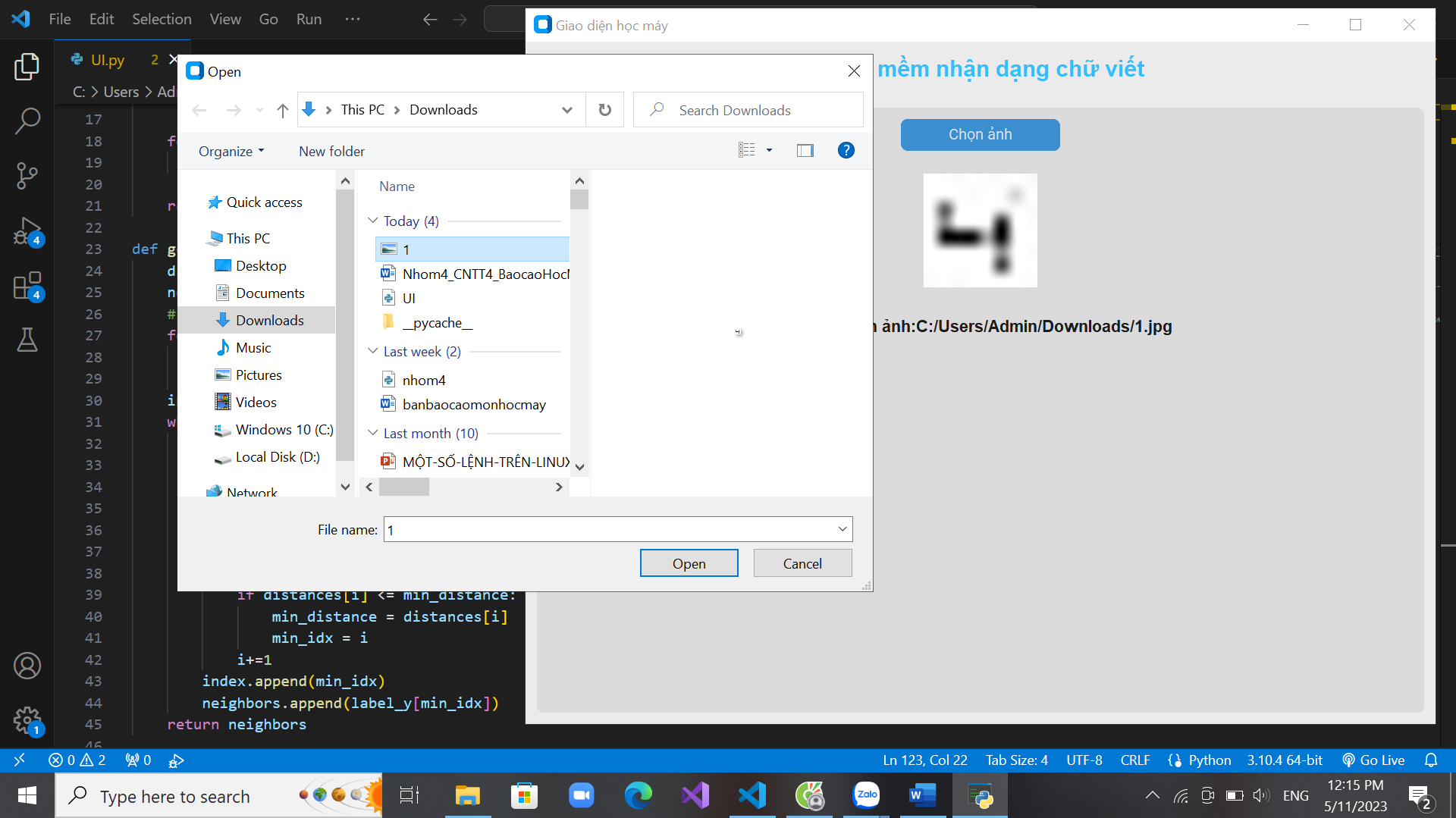
Sau đó khởi động GUI.

Các bước sử dụng chương trình như sau:

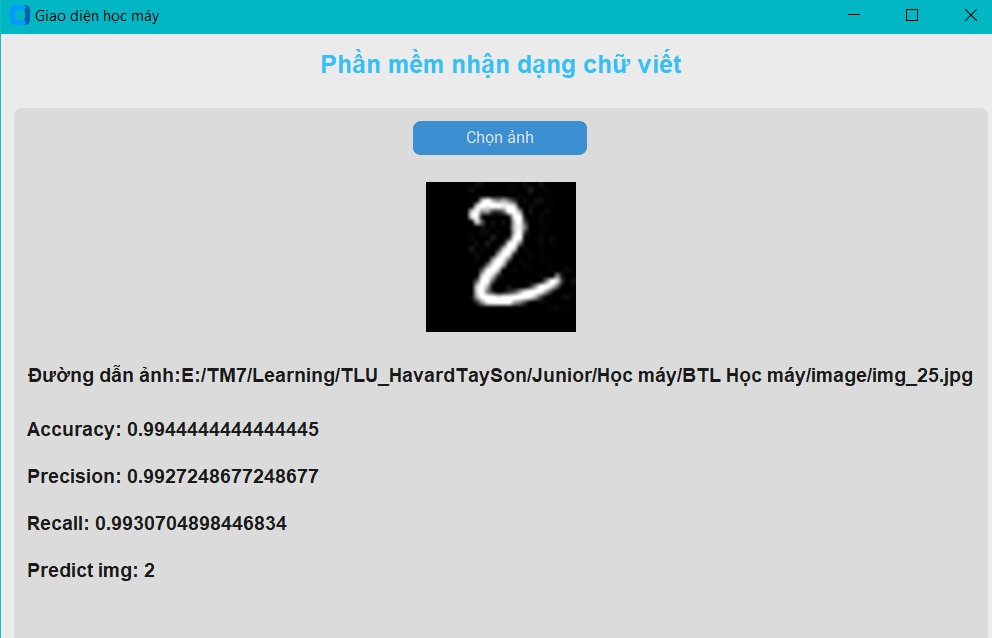
1. Chạy chương trình.



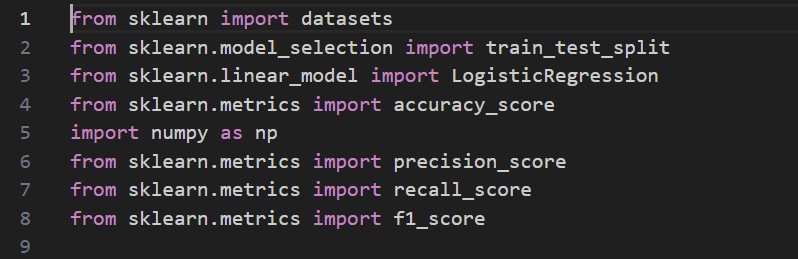
1. Nhấp vào nút "Chọn ảnh" và chọn tệp hình ảnh chứa chữ số mà bạn muốn nhận dạng.



1. Nhấp vào nút "Predict img" để xem kết quả dự đoán.

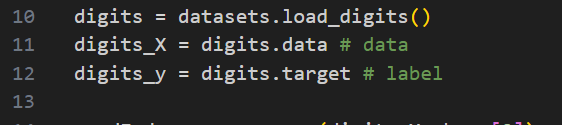


**3.2 - Xây dựng bằng thuật toán Logistics Regrestion.**

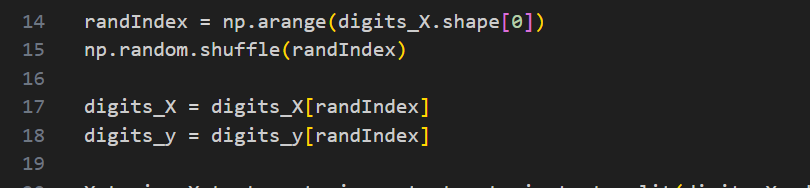


Đầu tiên,khai báo các thư viện cần thiết, bao gồm:

* **random**: để xáo trộn dữ liệu
* **sklearn.datasets**: để tải dữ liệu số
* **sklearn.model\_selection**: để chia dữ liệu thành tập huấn luyện và tập kiểm tra
* **sklearn.linear\_model**: để tạo mô hình hồi quy logistic
* **sklearn.metrics**: để đo lường hiệu suất của mô hình



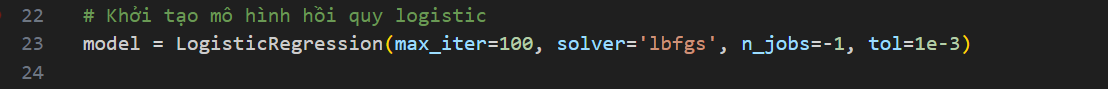
Load dữ liệu và lưu trữ dữ liệu và nhãn tương ứng vào các biến **digits\_X** và **digits\_y.**



Để xáo trộn dữ liệu, cần tạo một mảng **randIndex** chứa các chỉ số của dữ liệu theo thứ tự ngẫu nhiên. Sau đó,sử dụng hàm **np.random.shuffle()** để xáo trộn mảng này. Cuối cùng,sử dụng mảng **randIndex** để sắp xếp lại dữ liệu và nhãn theo thứ tự ngẫu nhiên.

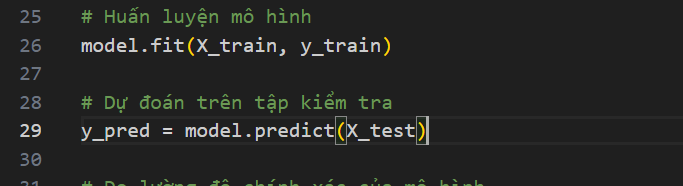


Chia dữ liệu thành tập huấn luyện và tập kiểm tra theo tỷ lệ 75:25. Điều này có nghĩa là 75% dữ liệu sẽ được sử dụng để huấn luyện mô hình và 25% dữ liệu còn lại sẽ được sử dụng để kiểm tra mô hình.

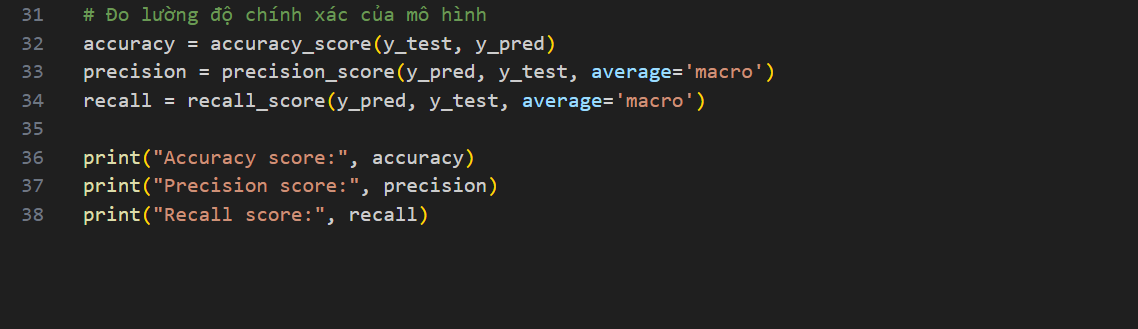


Khởi tạo mô hình hồi quy logistic với các tham số sau:

* **max\_iter**: số lần lặp tối đa của thuật toán huấn luyện
* **solver**: thuật toán huấn luyện được sử dụng
* **n\_jobs**: số luồng được sử dụng để huấn luyện mô hình
* **tol**: sai số tối đa được chấp nhận

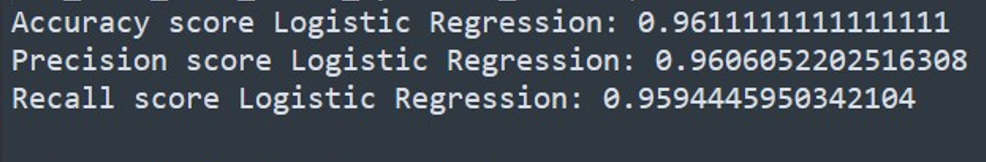


Huấn luyện mô hình trên tập huyến luyện.Và sau đó mô hình sẽ dự đoán trên tập kiểm tra.



Sử dụng các hàm **accuracy\_score()**, **precision\_score()**, và **recall\_score()** để đo lường độ chính xác, độ chính xác và độ nhạy của mô hình.Và in kết quả ra màn hình.

Kết quả chạy chương trình:

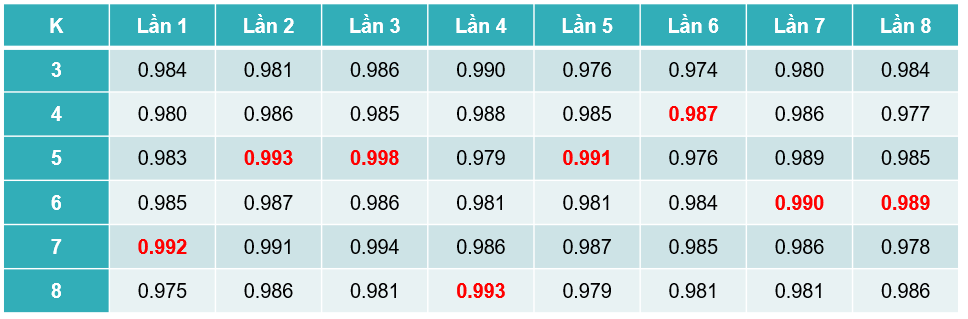


**4. Kết luận bài toán và hướng phát triển.**

Với cùng một bộ dữ liệu,chúng em đã thu được độ đo chính xác của 2 thuật toán KNN và Logistic Regrestion ở dưới bảng sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuật toán** | **Accuracy** | **Precision** | **Recall** |
| **K-Nearest Neighbors** | 0.994 | 0.992 | 0.993 |
| **Logistic Regrestion** | 0.961 | 0.960 | 0.959 |

Nhìn qua,cả 2 thuật toán đều cho ra độ chính xác rất cao.Nhưng cũng qua bảng dữ liệu cho thấy thuật toán KNN tốt hơn với bài toán của chúng em.Và chúng em quyết định xây dựng và lấy thuật toán KNN làm code chính của bài toán.



Và từ dữ liệu bài toán và qua các bước phân tích,phương pháp K-nearest neighbors và sẽ tối ưu nhất với k = 5 (xét ).

Sau khi xây dựng và chạy chương trình nhiều lần với nhiều ảnh khác nhau,KNN đã cho ra dự đoán chính xác rất cao.

Tuy nhiên vẫn còn một số hạn chế nhất định:

+ Dữ liệu giới hạn với ảnh vuông và có đặc điểm là nền đen chữ trắng

+ Chỉ xét **k** trong khoảng nhất định, có thể với **k** lớn thì độ chính xác cao hơn

+ Chương trình code ít sự dụng thư viện trong Python và dữ liệu đầu vào lớn nên việc chạy chương trình mất khác nhiều thời gian.

**- Hướng phát triển :**

Qua bài toán này,chúng em mong muốn sẽ xây dựng nên những bài toán tốt ưu và mở rộng hơn như:

* + Dự đoán chữ cái và các kí tự khác.
  + Dự đoán dãy số,chữ,kí tự.
  + Dự đoán biển số xe qua các video và hình ảnh chụp.

**5. Lời cảm ơn.**

Trên đây là toàn bộ bài báo cáo của nhóm 5 về đề tài **“ Nhận diện chữ số viết bằng tay từ 0 – 9”**.

Chúng em xin gửi lời cảm ơn chân thành nhất đến thầy Trần Anh Đạt đã tận tình hướng dẫn, giúp đỡ chúng em hoàn thành bài báo cáo bài tập lớn này.

Trong quá trình thực hiện bài báo cáo, chúng em đã nhận được rất nhiều sự giúp đỡ từ thầy. Thầy đã chỉ dẫn chúng em cách chọn đề tài, xây dựng kế hoạch, thu thập dữ liệu, phân tích dữ liệu và viết báo cáo. Thầy cũng đã dành thời gian quý báu để giải đáp những thắc mắc của chúng em.

Nhờ sự hướng dẫn của thầy, nhóm đã có cơ hội học hỏi được nhiều kiến thức và kỹ năng mới. Chúng em hy vọng rằng bài báo cáo của nhóm đã đáp ứng được yêu cầu của quý thầy cô.

Cuối cùng, chúng em xin gửi lời cảm ơn đến gia đình và bạn bè đã luôn động viên, ủng hộ chúng em trong suốt quá trình thực hiện bài báo cáo này.

Một số nguồn tài liệu tham khảo:

1. [https://www.codelearn.io› sharing › thuat-toan-k-nearest-neighbors-knn](https://codelearn.io/sharing/thuat-toan-k-nearest-neighbors-knn)

2. <https://wru-my.sharepoint.com/:f:/g/personal/dat_trananh_tlu_edu_vn/EoJ1c_UgwYlNtVyMa5zUE-IB3m2CIUgLlHUngl7XKqlA9Q?e=JbimIx>

3. <https://viblo.asia/p/logistic-regression-bai-toan-co-ban-trong-machine-learning-924lJ4rzKPM>