**Phần 1: Tìm hiểu các phương pháp Optimizer trong huấn luyện mô hình học máy.**

1. **Gradient Descent (GD)**

* Định nghĩa: GD là một phương pháp tối ưu hóa cơ bản dùng để cập nhật trọng số bằng cách tính toán đạo hàm của hàm mất mát theo trọng số và cập nhật trọng số theo hướng ngược lại với gradient.
* Ưu điểm:

Dễ hiểu và dễ triển khai.

Đảm bảo hội tụ tại điểm cực tiểu toàn cục nếu hàm mất mát là lồi.

* Nhược điểm:

Hội tụ chậm và có thể không hiệu quả với các bộ dữ liệu lớn do cần tính toán gradient trên toàn bộ dữ liệu.

1. **Stochastic Gradient Descent (SGD)**

* Định nghĩa: SGD cập nhật trọng số sau mỗi mẫu hoặc một mini-batch nhỏ, điều này giúp cập nhật nhanh hơn và có khả năng thoát khỏi các điểm cực tiểu cục bộ.
* Ưu điểm:

Có thể xử lý dữ liệu lớn hơn và hội tụ nhanh hơn so với GD truyền thống.

Do tính ngẫu nhiên, SGD có khả năng thoát khỏi các điểm cực tiểu cục bộ.

* Nhược điểm:

Sự không ổn định trong quá trình hội tụ do sự biến đổi lớn của gradient.

Có thể cần nhiều epoch hơn để hội tụ.

1. **Momentum**

* Định nghĩa: Momentum là một biến thể của SGD, nó tích lũy gradient của các bước trước đó để đạt được hướng đi "quán tính" và giúp hội tụ nhanh hơn.
* Ưu điểm:

Hội tụ nhanh hơn SGD đơn thuần do "quán tính" giúp vượt qua các thung lũng.

Giảm sự biến động của gradient, dẫn đến quá trình học ổn định hơn.

* Nhược điểm:

Có thể gặp khó khăn khi điều chỉnh các hyperparameter như tỷ lệ học và momentum.

Có khả năng "vượt qua" điểm tối ưu.

1. **Adagrad**

* Định nghĩa: Adagrad điều chỉnh tỷ lệ học của mỗi tham số dựa trên tần suất của chúng xuất hiện trong tập dữ liệu, thích hợp cho việc xử lý dữ liệu thưa.
* Ưu điểm:

Tự động điều chỉnh tỷ lệ học và thích hợp cho các bài toán có dữ liệu thưa thớt.

Loại bỏ phần nào nhu cầu phải điều chỉnh tỷ lệ học thủ công.

* Nhược điểm:

Tỷ lệ học có thể giảm quá nhanh và mô hình có thể ngừng học trước khi đạt được điểm tối ưu.

1. **RMSprop**

* Định nghĩa: RMSprop là một biến thể của Adagrad, giải quyết vấn đề giảm nhanh của tỷ lệ học bằng cách sử dụng một trung bình di động của gradient vuông để điều chỉnh tỷ lệ học.
* Ưu điểm:

Tự động điều chỉnh tỷ lệ học và nói chung hoạt động tốt trong nhiều bài toán.

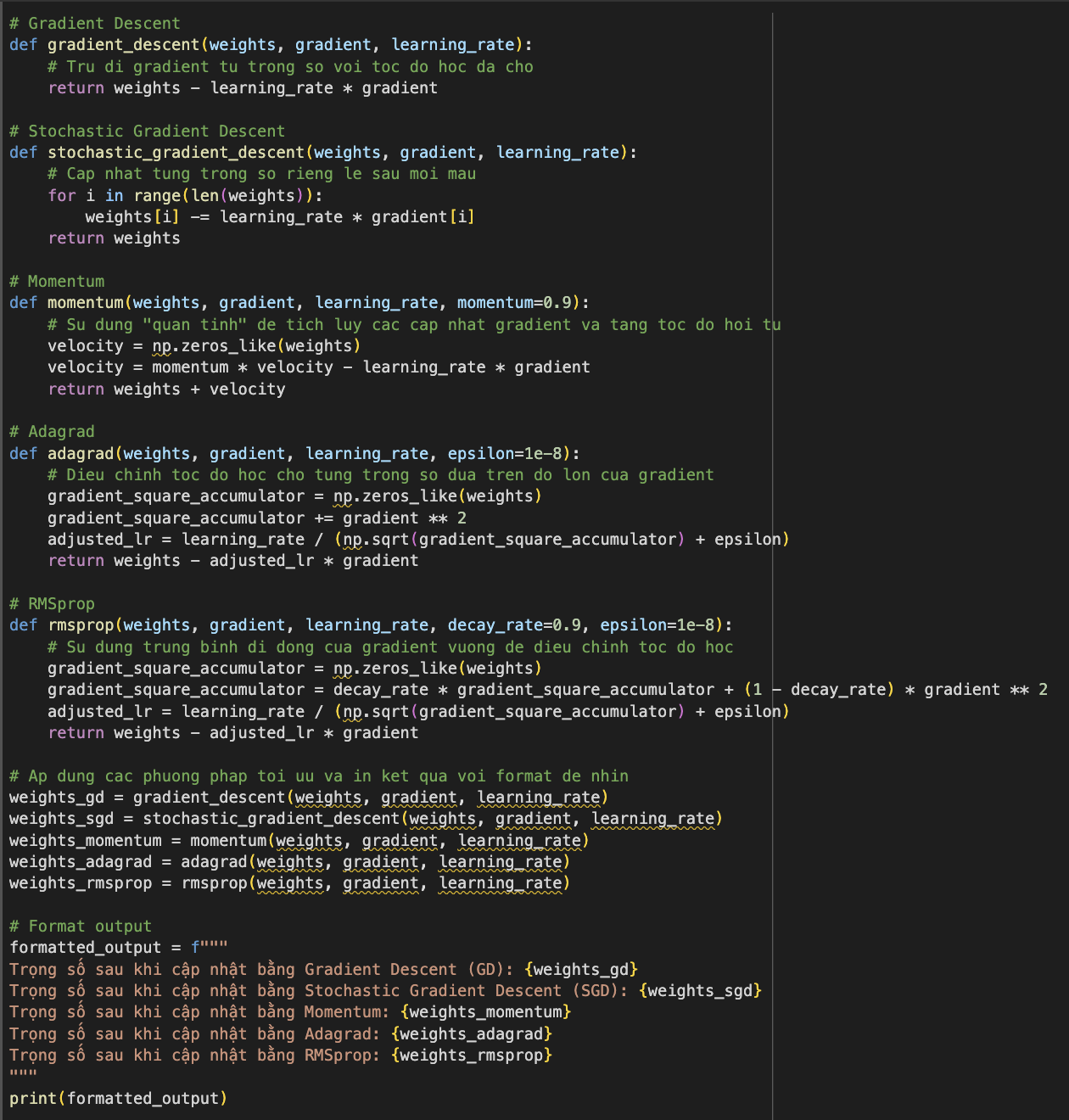
thiểu vấn đề giảm quá nhanh của tỷ lệ học so với Adagrad.

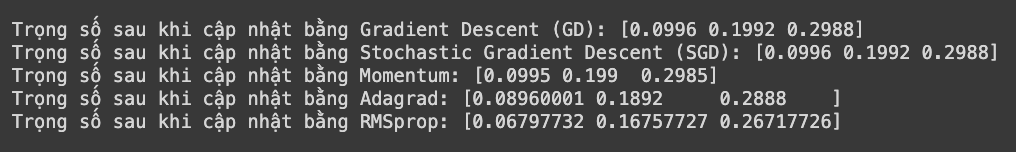
* Nhược điểm:

Vẫn có thể gặp khó khăn trong việc chọn các hyperparameter tốt nhất.

Có thể không hiệu quả như các phương pháp tối ưu hóa tiên tiến khác trong một số trường hợp.

1. **Code minh hoạ cho các mô hình trên**



Kết quả:

Nhận xét:

**Gradient Descent (GD) và Stochastic Gradient Descent (SGD):** Cả hai phương pháp đều cung cấp cùng một kết quả khi mỗi cập nhật được thực hiện trên toàn bộ dữ liệu hoặc một mẫu duy nhất. Nhưng trong trường hợp này, chưa thấy được sự khác biệt giữa GD và SGD do cách chúng được mô phỏng.

**Momentum:** Momentum giúp tiếp cận điểm tối ưu nhanh hơn bằng cách giảm bớt sự biến động và tăng tốc độ hội tụ. Điều này là do phương pháp này tích lũy các cập nhật từ các bước trước đó, giúp "đẩy" quá trình tối ưu hóa vượt qua các thung lũng.

**Adagrad:** Có sự điều chỉnh tỷ lệ học đáng kể cho mỗi tham số, làm cho phương pháp này rất phù hợp với các bài toán có dữ liệu thưa. Tuy nhiên, tỷ lệ học có thể giảm quá nhanh, dẫn đến việc mô hình ngừng học sớm.

**RMSprop:** Cải thiện nhược điểm của Adagrad bằng cách điều chỉnh tỷ lệ học thông qua trung bình di động của gradient vuông, giúp tránh được vấn đề giảm tỷ lệ học quá nhanh và thích nghi tốt hơn với các bài toán khác nhau.

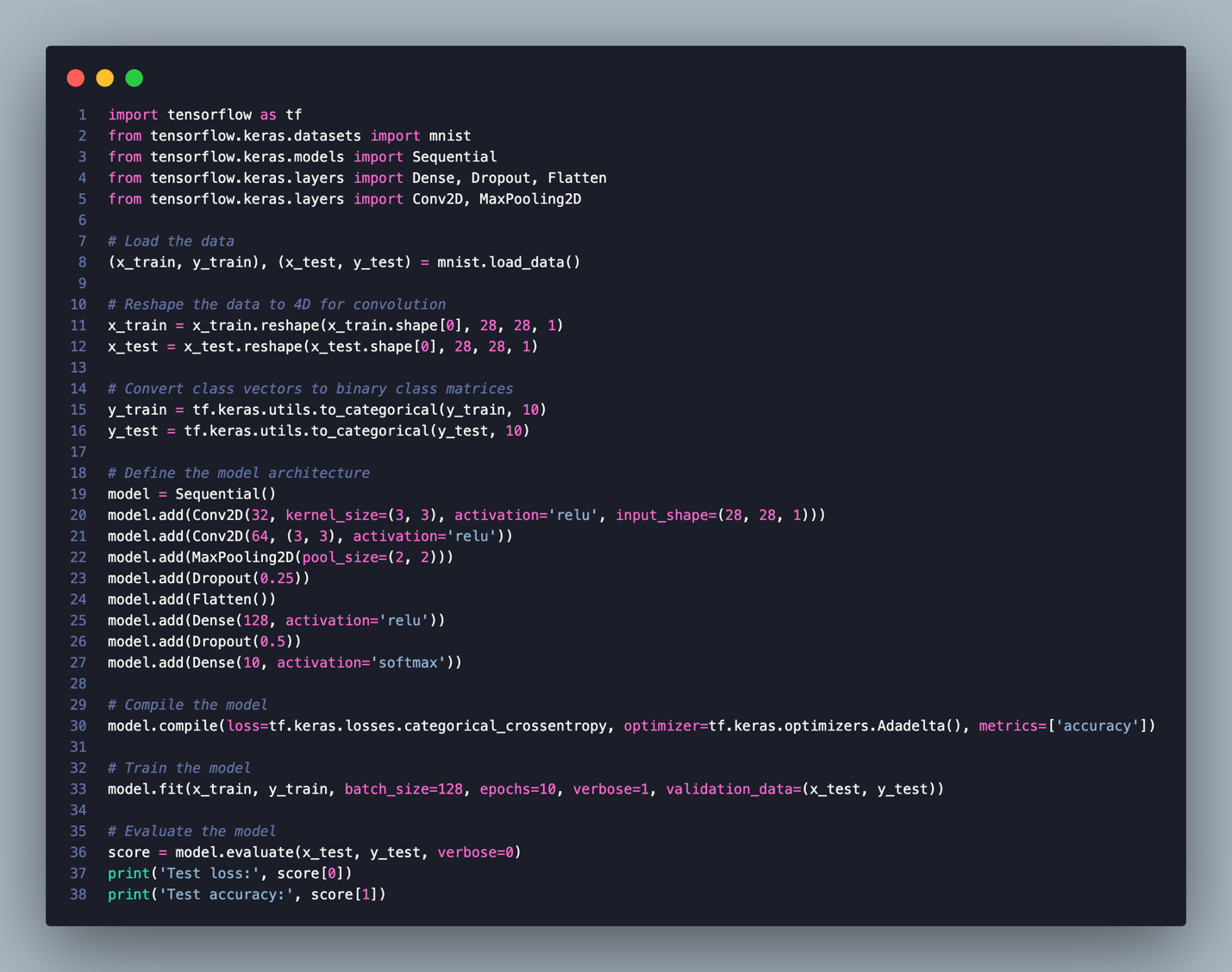
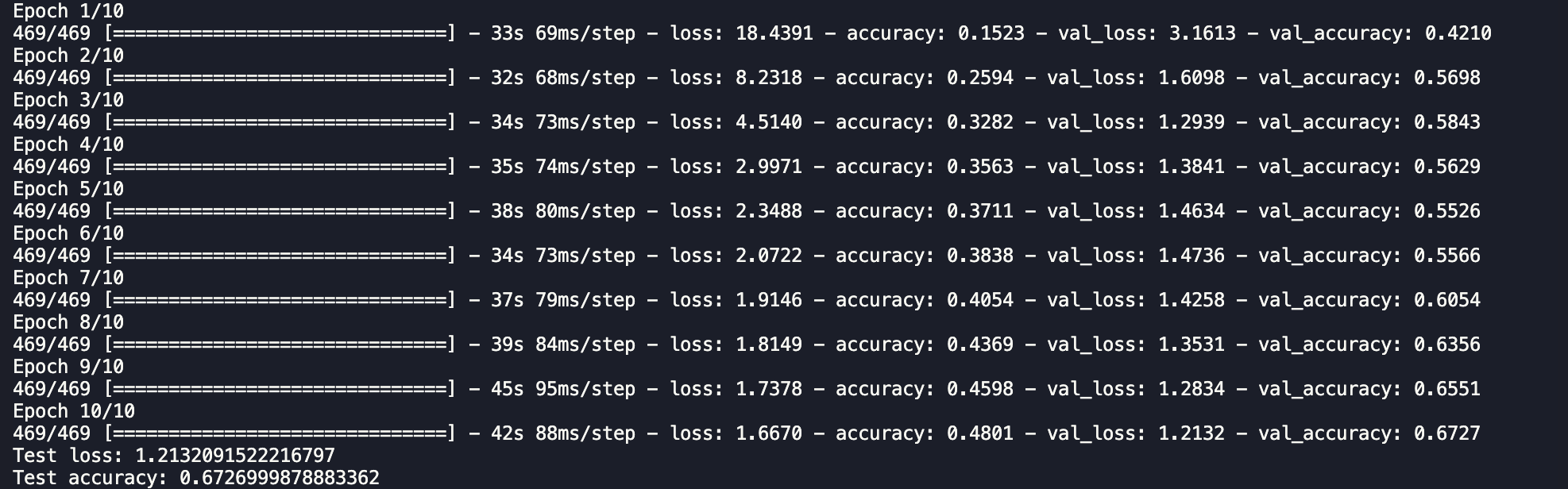
**Phần 2: Continual Learning và Test Production khi xây dựng giải pháp học máy giải quyết bài toán**

1. **Continual Learrning**

Continual Learning (CL) là một phương pháp học máy, cho phép hệ thống tự động cập nhật và mở rộng kiến thức mà không cần phải thay đổi toàn bộ hệ thống. CL được thực hiện bằng cách thu thập và xử lý dữ liệu liên tục từ các nguồn khác nhau, thông qua việc cập nhật bộ phận học của hệ thống.

**Minh hoạ khi xây dựng Continual Learning trong một bài toán học máy**

Đoạn code sau sử dụng thư viện TensorFlow để tạo mô hình Convolutional Neural Network (CNN) và đánh giá khả năng chính xác của nó trên bộ dữ liệu MNIST

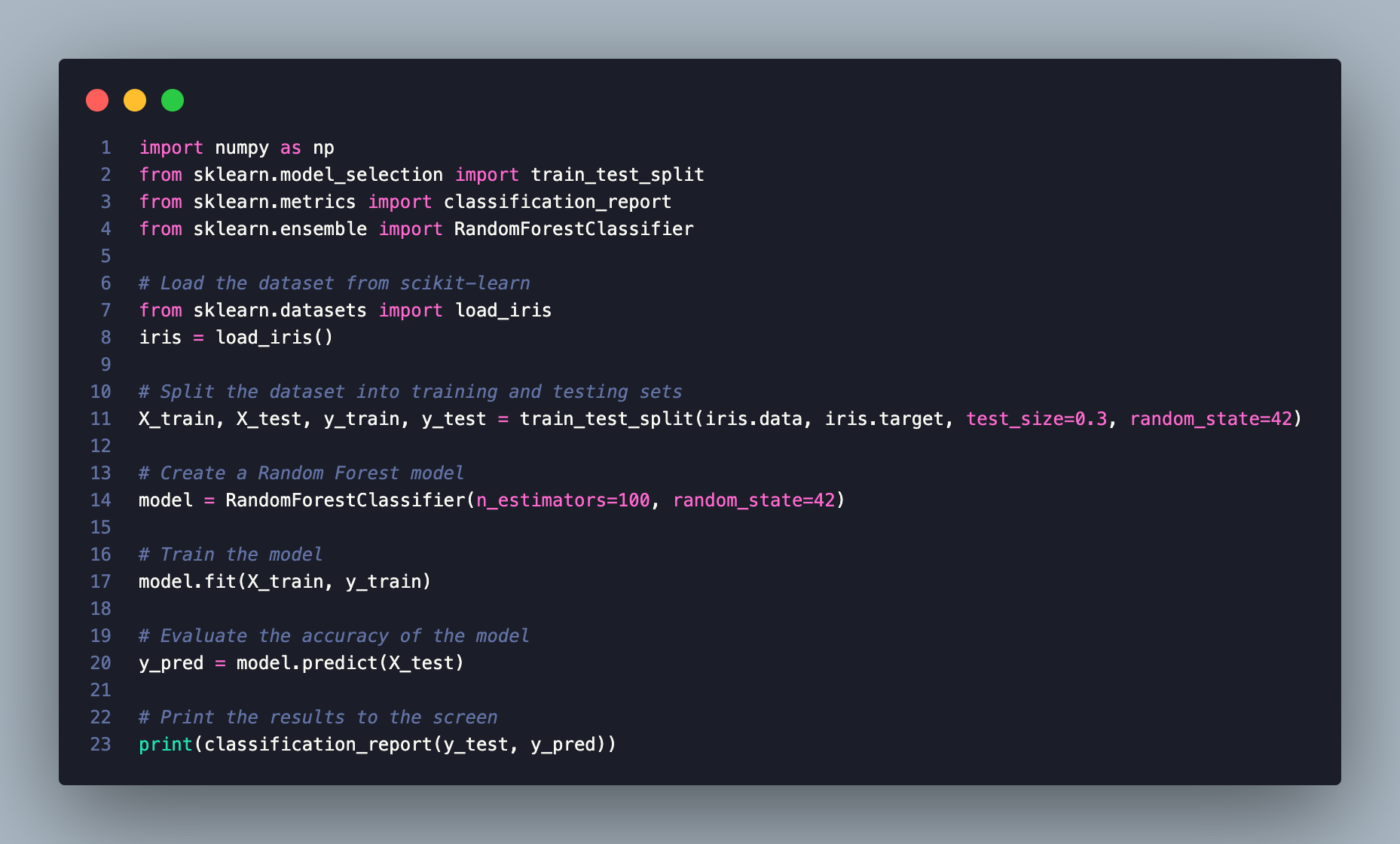
**Kết quả:**

1. **Test Production:**

Test Production (tương đương với việc xác thực thông minh trong xử lý ngôn ngữ tự nhiên) là một kỹ thuật xác định chính xác vị trí nhỏ nhất trong một chuỗi đầu vào nơi những sự biến đổi xuất hiện.

**Minh hoạ khi xây dựng Test Production trong một bài toán học máy**

Đoạn code sau sử dụng bộ dữ liệu Iris từ scikit-learn để minh họa cách sử dụng thư viện này.



Kết quả:

