**Kiểm thử tự động là gì?**

Trong kiểm thử phần mềm, kiểm thử tự động là việc sử dụng phần mềm tách biệt với phần mềm đang được kiểm tra để kiểm soát việc thực hiện ca kiểm thử và so sánh kết quả thực tế với kết quả dự đoán [2]. Nó được sử dụng nhiều lần cho kiểm thử hồi quy. Người kiểm thử tạo các tập lệnh và chạy tập lệnh, về cơ bản nó là sự tự động hóa ca kiểm thử thủ công. Kiểm thử tự động cũng được sử dụng để kiểm tra khả năng tải và hiệu năng của ứng dụng. Mục tiêu chính của kiểm thử tự động là giảm kiểm thử thủ công, tăng hiệu quả kiểm thử và giá trị của phần mềm.

**Trí tuệ nhân tạo là gì?**

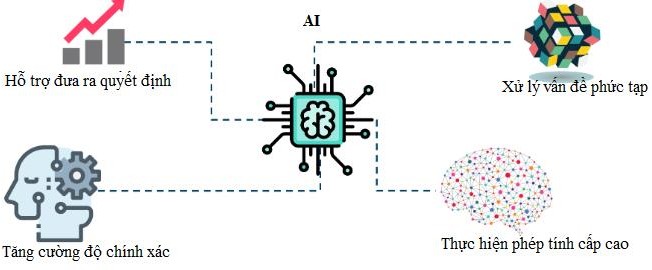
Trong khoa học máy tính, trí tuệ nhân tạo (tên tiếng anh là Artificial Intelligence - AI ) là trí thông minh được thể hiện bằng máy móc. Nó xử lý những phép tính cấp cao, vấn đề phức tạp và hỗ trợ hệ thống đưa ra những quyết định nhằm tăng cường độ chính xác. Thông thường, thuật ngữ trí tuệ nhân tạo được dùng để mô tả máy móc bắt chước nhận thức của con người như học tập và giải quyết vấn đề. Ví dụ như các ứng dụng để giúp tài xế tìm các tuyến đường hiệu quả, dữ liệu thời gian thực của các tuyến đường được truyền về và phân tích, học hỏi để ứng dụng đưa ra kết luận về tuyến đường.

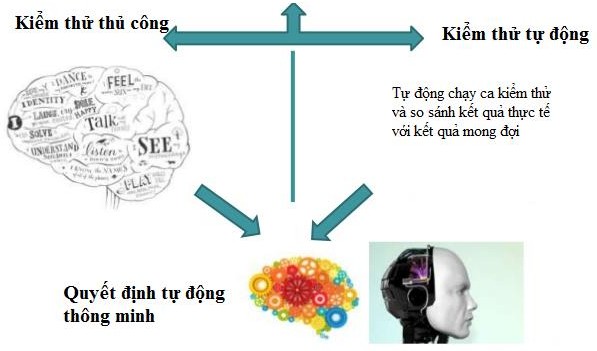
Học máy là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Ví dụ như các máy có thể "học" cách phân loại thư điện tử xem có phải thư rác hay không và tự động xếp thư vào thư mục tương ứng. Học máy rất gần với suy diễn thống kê tuy có khác nhau về thuật ngữ. Học máy nghiên cứu cách thức để mô hình hóa bài toán cho phép máy tính tự động hiểu, xử lý và học từ dữ liệu để thực thi nhiệm vụ được giao cũng như cách đánh giá giúp tăng tính hiệu quả.

**Xu thế sử dụng học máy trong kiểm thử tự động**

Việc sử dụng trí tuệ nhân tạo trong phát triển phần mềm vẫn còn ở mức độ tự chủ thấp hơn nhiều so với các lĩnh vực phát triển hơn như hệ thống tự lái hoặc điều khiển hỗ trợ giọng nói. Ứng dụng của AI trong các công cụ kiểm thử phần mềm tập trung vào việc làm cho vòng đời phát triển phần mềm trở nên dễ dàng hơn. Hình sau chỉ ra việc trí tuệ nhân

tạo sẽ học việc quan sát và đưa ra quyết định của con người trong kiểm thử thủ công để tự động đưa ra quyết .Học máy có thể được sử dụng để giúp tự động hóa và giảm số lượng các nhiệm vụ thủ công trong phát triển và thử nghiệm.

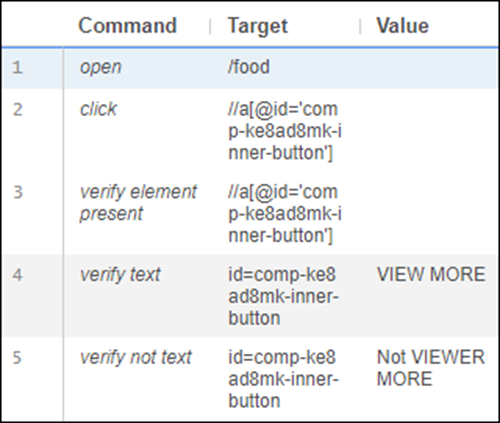




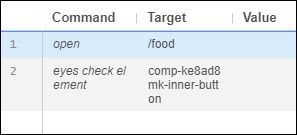
Người kiểm thử có sự thông minh và khả năng quan sát trạng thái kết quả, máy kiểm thử có hiệu suất cao. Để nâng cao hiệu quả và giảm chi phí chất lượng, cần cải thiện kiểm thử phần mềm bằng cách đưa ra hành vi kiểm thử tự động thông minh thứ có khả năng bắt chước hình vi con người. Trong kiểm thử phần mềm, một vấn đề được phát hiện ra

sau quá trình áp dụng đầu vào ứng dụng, so sánh kết quả với mong đợi đầu ra. Nó chính xác với những gì học máy làm. Một tập đầu vào được cung cấp cho một thuật toán đào tạo. Trong quá trình học tập có giám sát, câu trả lời đúng cho mỗi tập đầu vào cũng được cung cấp cho thuật toán đào tạo. Công việc của hệ thống học máy là lặp đi lặp lại sự cấu hình lại “nội bộ của bộ não”, ngày càng tốt hơn trong việc cung cấp các câu trả lời đúng dựa trên tập đầu vào đã được cung cấp. Do đó, những hệ thống ML đã và đang được nghiên cứu phát triển đều cung cấp những lợi ích trực tiếp cho việc tự động hóa kiểm thử phần mềm.

Với kiểm thử tự động thông thường, con người viết các dòng lệnh để thực hiện ca kiểm thử tự động, sau đó công cụ sẽ thực hiện chạy và so sánh đưa ra kết quả dựa vào các dòng lệnh đã được viết; với kiểm thử tự động có ứng dụng trí tuệ nhân tạo, việc tạo ca kiểm thử và thực hiện so sánh kết quả đều được công cụ thực hiện. Ví dụ với ca kiểm thử xác minh phẩn tử “VIEW MORE”. Đoạn lệnh thực hiện với công cụ Selenium bình thường, khi ca kiểm thử được chạy, việc so sánh kết quả chỉ dừng ở thực hiện đúng nội dung “verify text” and “verify not text”; nếu muốn so sánh càng chính xác, càng nhiều câu lệnh cần được viết. Nhưng với công cụ có trí tuệ nhân tạo (ví dụ Applitools, Testim.io, Testcomplete, Mabl…), dòng lệnh được rút gọn, nhưng sự so sánh đưa ra kết quả lại dựa trên rất nhiều tiêu chí như so sánh màu sắc, nội dung, sự sai khác đến từng đơn vị điểm ảnh. Dòng lệnh với Applitools. Với những trường hợp phức tạp như dữ liệu biểu đồ bảng, với các công cụ kiểm thử tự động đơn thuần, việc viết ca kiểm thử cũng rất phức tạp, nhưng nó được giải quyết chỉ bằng một vài dòng lệnh với công cụ có ứng dụng trí tuệ nhân tạo (ví dụ như Testcomplete).

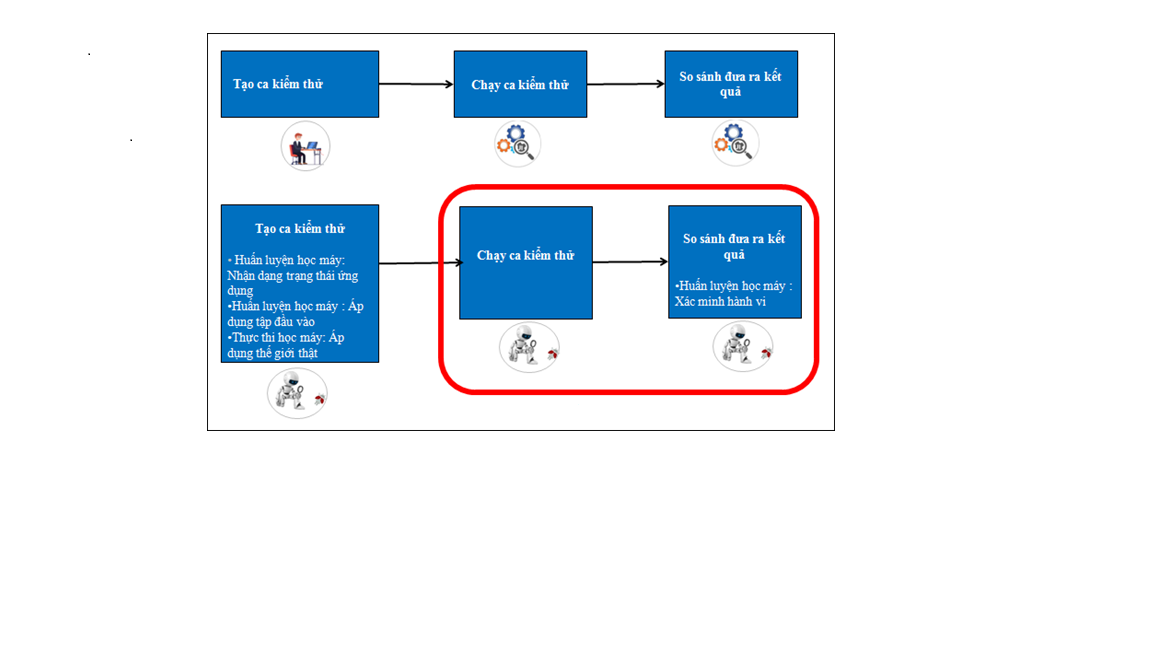


bộ lệnh Selenium



bộ lệnh Applitools

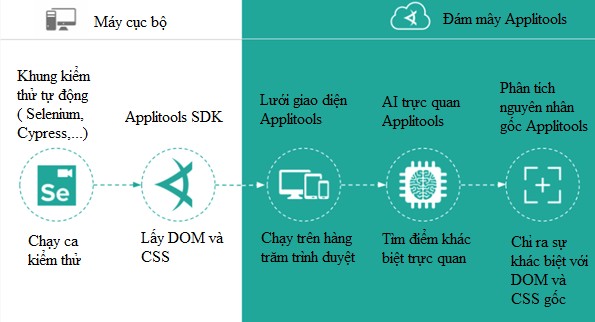
Trí tuệ nhân tạo và học máy được ứng dụng trong các lĩnh vực của kiểm thử, ví dụ như kiểm thử xác minh giao diện, kiểm thử API, kiểm thử bảo mật…. Các công cụ có ứng dụng trí tuệ nhân tạo cũng có thể tự sinh ca kiểm thử, con người chỉ cần trỏ cho công cụ biết ứng dụng cần sinh ca kiểm thử ở đâu. Khi một đoạn mã thay đổi, ML cũng có khả năng tính toán để chỉ ra chính xác số lượng ca kiểm thử nhỏ nhất để kiểm tra việc thay đổi đoạn mã. ML cũng được sử dụng để tạo ca kiểm thử đáng tin cậy hơn, nó tìm hiểu và quan sát cách ứng dụng thay đổi, sau đó tự đưa ra quyết định trong thời gian chạy về việc sử dụng bộ định vị nào để xác định phần tử.



**Applitools** tập trung vào các kiểm thử trực quan, xác minh giao diện xuất hiện như dự định. Nó cung cấp các API khác nhau cho các ngôn ngữ lập trình và các khung (tên tiếng anh là: framework) để tạo kiểm thử trực quan cho các thành phần GUI đơn lẻ hoặc toàn bộ màn hình.

Trong quá trình thực thi kiểm thử, mọi kiểm tra thông qua Applitools API tạo ra một ảnh chụp màn hình và tải lên một dịch vụ chuyên dụng (máy chủ đôi mắt) để so sánh. Tại đây, các thuật toán thị giác máy tính (CV) được sử dụng chỉ để báo cáo sự khác biệt có thể nhận thấy. Sau đó, quản lý kiểm thử đôi mắt có thể được sử dụng để xem xét chi tiết kết quả kiểm thử. Các thay đổi được phê duyệt, từ chối hoặc bỏ qua, trong đó các thay đổi tương tự được nhóm lại với nhau để tăng tốc quá trình bảo trì.

**Applitools hoạt động như thế nào?**

* Khung kiểm thử tự động: cài đặt trên máy cục bộ của người dùng để tạo, chạy ca kiểm thử
* Applitools SDK: cài đặt trên máy cục bộ của người dùng, định nghĩa các nguyên tắc DOM, CSS của các phần tử khi chạy ca kịch bản.
* Lưới giao diện Applitools (tên tiếng anh là Applitools Visual Gird): nằm trên đám mây Applitools, ca kiểm thử được chạy trên lưới giao diện.
* AI trực quan Applitools (tên tiếng anh là Applitools Visual AI) hay còn gọi là “Eyes”: nằm trên đám mây Applitools, so sánh sự khác biệt về giao diện và thông báo kết quả.
* Phân tích nguyên nhân góc Applitools (tên tiếng anh là Applitools Root Cause Analysis): nằm trên đám mây Applitools, quản lý kết quả so sánh.

Khi ca kiểm thử được chạy lần đầu tiên trên ứng dụng, máy chủ AI chỉ lưu trữ ảnh chụp màn hình dưới dạng ảnh cơ sở (tên tiếng anh :baseline). Khi một ca kiểm thử tương tự được chạy lần tiếp theo, tập ảnh chụp màn hình mới được gọi là ảnh kiểm tra (tên tiếng anh: checkpoint). Eyes sẽ so sánh ảnh cơ sở và ảnh kiểm tra để đưa ra quyết định.

***AI trong Applitools***

Thị giác máy tính là công nghệ được sử dụng cho Eyes. AI được xử lý trên đám mây Applitools. Các thuật toán chỉ báo cáo sự khách biệt về giao điện người dùng có ý nghĩa và có thể cảm nhận được bằng mắt người.

Công nghệ của Applitools cho phép xác nhận rằng cấu trúc vẫn giống nhau, mặc dù nội dung đã thay đổi. Thay vì có một liên kết tĩnh giữa mọi ảnh kiểm tra và ảnh cơ sở tương ứng, Eyes so sánh một chuỗi các hình ảnh kiểm tra với một chuỗi các hình ảnh cơ sở, cho phép các điểm kiểm tra có thể được thêm hoặc xóa. Sau đó, giống như một chương trình "khác biệt" bằng văn bản, Eyes tìm thấy hình ảnh đã được thêm hoặc xóa cũng như hình ảnh có hình ảnh đường cơ sở tương ứng và khớp hoặc không khớp. Eyes sau đó cho phép bạn xác minh cả hai điểm kiểm tra hiện có và đảm bảo rằng tập hợp các hình ảnh cơ sở

được cập nhật. Bởi vì AI không biết sự khác biệt là do tính năng mới hay lỗi thực sự, do đó nó sẽ đợi con người đánh dấu thủ công là thông qua hay thất bại.

Các bước so sánh và kết quả kiểm tra: Khi ca kiểm thử được thực thi, Eyes xác minh toàn bộ chuỗi ảnh kiểm tra đối với chuỗi ảnh cơ sở. Kết quả kiểm tra bao gồm một chuỗi các bước, mỗi bước biểu thị một điểm kiểm tra bị thiếu, một điểm kiểm tra mới, một điểm kiểm tra phù hợp với đối tác cơ sở của nó hoặc một điểm khác với nó.

Ví dụ, chuỗi ảnh cơ sở bao gồm 5 ảnh:



Applitools - ví dụ ảnh cơ sở

Và chuỗi ảnh kiểm tra:



Applitools - ví dụ ảnh kiểm tra

Sự thay đổi của chuỗi ảnh kiểm tra: ảnh B bị xóa, ảnh C bị thay bởi ảnh C1, ảnh F và G được thêm mới, ảnh A, D, E không thay đổi. Eyes sẽ phát hiện sự giống khác giữa hai chuỗi ảnh:

* Giống: Eyes tìm thấy ảnh cơ sở và ảnh kiểm tra giống nhau.
* Khác: Eyes tìm thấy ảnh cơ sở và ảnh kiểm tra về cơ bản là khớp nhau nhưng có một phần khác biệt.
* Thiếu: Eyes không tìm thấy ảnh kiểm tra tương ứng với ảnh cơ sở.
* Mới: Eyes phát hiện ra ảnh cơ sở mới.