Script thuyết trình httm

Việc định hình một bài toán học máy thường bao gồm nhiều cuộc thảo luận chi tiết với các bên liên quan. Dưới đây là những câu hỏi bạn nên luôn nhớ trong đầu:

* **Dữ liệu đầu vào của bạn là gì? Bạn đang cố gắng dự đoán điều gì?** Bạn chỉ có thể học để dự đoán một điều gì đó nếu có dữ liệu huấn luyện sẵn có: ví dụ, bạn chỉ có thể học cách phân loại cảm xúc trong các bài đánh giá phim nếu bạn có cả bài đánh giá phim và nhãn cảm xúc. Do đó, khả năng có dữ liệu là yếu tố giới hạn chính ở giai đoạn này. Trong nhiều trường hợp, bạn sẽ phải tự thu thập và chú thích các bộ dữ liệu mới (chúng tôi sẽ đề cập đến điều này trong phần tiếp theo).
* **Bạn đang đối mặt với loại bài toán học máy nào?** Là phân loại nhị phân? Phân loại đa lớp? Hồi quy số học? Hồi quy véc-tơ? Phân loại đa lớp, đa nhãn? Phân đoạn ảnh? Xếp hạng? Hay một thứ khác, như phân cụm, sinh dữ liệu, hoặc học tăng cường? Trong một số trường hợp, có thể học máy không phải là cách tốt nhất để giải quyết dữ liệu, và bạn nên sử dụng một phương pháp khác, như phân tích thống kê cổ điển.
  + Dự án tìm kiếm ảnh là một bài toán phân loại đa lớp, đa nhãn.
  + Dự án phát hiện thư rác là một bài toán phân loại nhị phân. Nếu bạn coi "nội dung xúc phạm" là một lớp riêng biệt, thì đó là bài toán phân loại ba lớp.
  + Dự án gợi ý nhạc thực tế lại được giải quyết tốt hơn không phải bằng học sâu, mà là thông qua phân tích ma trận (lọc cộng tác).
  + Dự án phát hiện gian lận thẻ tín dụng là một bài toán phân loại nhị phân.
  + Dự án dự đoán tỷ lệ nhấp chuột (click-through rate) là một bài toán hồi quy số học.
  + Phát hiện cookie bất thường là một bài toán phân loại nhị phân, nhưng nó cũng sẽ yêu cầu một mô hình phát hiện đối tượng ở giai đoạn đầu tiên để cắt đúng cookie trong hình ảnh gốc. Lưu ý rằng các kỹ thuật học máy được gọi là "phát hiện bất thường" sẽ không phù hợp trong trường hợp này!
  + Dự án tìm kiếm các địa điểm khảo cổ mới từ ảnh vệ tinh là một bài toán xếp hạng tương tự hình ảnh: bạn cần tìm những hình ảnh mới giống nhất với các địa điểm khảo cổ đã biết.
* **Giải pháp hiện có là gì?** Có thể khách hàng của bạn đã có một thuật toán thủ công để xử lý việc lọc thư rác hoặc phát hiện gian lận thẻ tín dụng, với nhiều câu lệnh if lồng nhau. Có thể một con người hiện đang chịu trách nhiệm xử lý thủ công quy trình đang xem xét—giám sát băng chuyền tại nhà máy cookie và loại bỏ thủ công các cookie xấu, hoặc tạo ra danh sách phát các bài hát gợi ý để gửi đến người dùng đã thích một nghệ sĩ cụ thể. Bạn nên hiểu rõ những hệ thống hiện có và cách chúng hoạt động.
* **Có những ràng buộc đặc biệt nào bạn cần phải xử lý không?** Ví dụ, bạn có thể phát hiện ra rằng ứng dụng mà bạn đang xây dựng hệ thống phát hiện thư rác có mã hóa đầu cuối nghiêm ngặt, vì vậy mô hình phát hiện thư rác sẽ phải chạy trên điện thoại của người dùng cuối và phải được huấn luyện trên một bộ dữ liệu bên ngoài. Có thể mô hình lọc cookie có các ràng buộc về độ trễ đến mức phải chạy trên thiết bị nhúng tại nhà máy thay vì trên một máy chủ từ xa. Bạn cần hiểu rõ ngữ cảnh đầy đủ mà công việc của bạn sẽ áp dụng.

Khi bạn đã hoàn thành nghiên cứu, bạn sẽ biết đầu vào của mình là gì, mục tiêu của bạn là gì, và bài toán sẽ thuộc loại học máy nào. Hãy nhận thức rõ những giả thuyết bạn đang đưa ra ở giai đoạn này:

* **Bạn giả thuyết rằng các mục tiêu của bạn có thể được dự đoán dựa trên các đầu vào của bạn.**
* **Bạn giả thuyết rằng dữ liệu hiện có (hoặc dữ liệu bạn sắp thu thập) đủ thông tin để học mối quan hệ giữa đầu vào và mục tiêu.**

Cho đến khi bạn có một mô hình hoạt động, đây chỉ là những giả thuyết, chờ được xác nhận hoặc bác bỏ. Không phải tất cả các vấn đề đều có thể được giải quyết bằng học máy; chỉ vì bạn đã tập hợp được các ví dụ về đầu vào X và mục tiêu Y không có nghĩa là X chứa đủ thông tin để dự đoán Y. Ví dụ, nếu bạn đang cố gắng dự đoán chuyển động của một cổ phiếu trên thị trường chứng khoán dựa trên lịch sử giá gần đây của nó, bạn sẽ khó thành công, vì lịch sử giá không chứa nhiều thông tin dự đoán.

**Quá Trình Chú Thích Dữ Liệu Của Bạn Sẽ Quyết Định Chất Lượng Của Các Mục Tiêu, Và Từ Đó Ảnh Hưởng Đến Chất Lượng Của Mô Hình**

Hãy cân nhắc kỹ lưỡng các lựa chọn mà bạn có trong quá trình chú thích dữ liệu:

* **Có nên tự chú thích dữ liệu không?**
* **Có nên sử dụng nền tảng crowdsourcing như Mechanical Turk để thu thập nhãn không?**
* **Có nên sử dụng dịch vụ của một công ty chuyên cung cấp dịch vụ gắn nhãn dữ liệu không?**

Thuê ngoài có thể giúp bạn tiết kiệm thời gian và chi phí, nhưng điều này sẽ làm bạn mất kiểm soát. Sử dụng dịch vụ như Mechanical Turk có thể rẻ và dễ mở rộng, nhưng chú thích của bạn có thể sẽ bị nhiễu hoặc không chính xác.

Để chọn lựa phương án tốt nhất, hãy xem xét các ràng buộc bạn đang làm việc với:

* **Liệu người chú thích dữ liệu có cần phải là chuyên gia trong lĩnh vực, hay bất kỳ ai cũng có thể chú thích dữ liệu?** Ví dụ, các nhãn cho bài toán phân loại hình ảnh mèo-với-chó có thể được chọn bởi bất kỳ ai, nhưng việc phân loại giống chó yêu cầu kiến thức chuyên môn. Trong khi đó, chú thích các ảnh chụp CT về các vết gãy xương gần như đòi hỏi một bằng cấp y tế.
* **Nếu việc chú thích dữ liệu yêu cầu kiến thức chuyên môn, liệu bạn có thể huấn luyện người khác làm điều đó không? Nếu không, làm thế nào để bạn có thể tiếp cận các chuyên gia có liên quan?**
* **Bạn có hiểu cách mà các chuyên gia đưa ra các nhãn không?** Nếu bạn không hiểu, bạn sẽ phải coi dữ liệu của mình như một "hộp đen", và bạn sẽ không thể thực hiện kỹ thuật tạo đặc trưng thủ công (manual feature engineering)—điều này không phải là yếu tố quan trọng, nhưng nó có thể là một hạn chế.

Nếu bạn quyết định tự chú thích dữ liệu trong công ty, hãy tự hỏi bạn sẽ sử dụng phần mềm nào để ghi lại các nhãn. Có thể bạn sẽ phải tự phát triển phần mềm đó. Phần mềm chú thích dữ liệu hiệu quả sẽ giúp bạn tiết kiệm rất nhiều thời gian, vì vậy đáng để đầu tư vào nó ngay từ đầu dự án.

**CẨN THẬN VỚI DỮ LIỆU KHÔNG ĐẠI DIỆN**

Mô hình học máy chỉ có thể hiểu được các đầu vào tương tự như những gì chúng đã thấy trước đây. Do đó, điều quan trọng là dữ liệu sử dụng để huấn luyện phải đại diện cho dữ liệu sản xuất. Mối quan tâm này nên là nền tảng của tất cả công việc thu thập dữ liệu của bạn.

Giả sử bạn đang phát triển một ứng dụng cho phép người dùng chụp ảnh một đĩa thức ăn để tìm ra tên món ăn. Bạn huấn luyện một mô hình sử dụng những bức ảnh từ một mạng xã hội chia sẻ hình ảnh nổi tiếng với các tín đồ ẩm thực. Tuy nhiên, khi triển khai, bạn bắt đầu nhận phản hồi từ người dùng tức giận: ứng dụng của bạn trả lời sai 8 lần trên 10. Điều gì đang xảy ra? Độ chính xác của bạn trên bộ dữ liệu kiểm tra vượt trên 90%! Một cái nhìn nhanh vào dữ liệu do người dùng tải lên cho thấy ảnh chụp món ăn từ các nhà hàng ngẫu nhiên, được chụp bằng điện thoại thông minh ngẫu nhiên, trông không giống như những bức ảnh chuyên nghiệp, có ánh sáng đẹp mà bạn đã huấn luyện mô hình trên đó: dữ liệu huấn luyện của bạn không đại diện cho dữ liệu sản xuất. Đây là một sai lầm nghiêm trọng—chào mừng bạn đến với "địa ngục học máy".

Nếu có thể, hãy thu thập dữ liệu trực tiếp từ môi trường mà mô hình của bạn sẽ được sử dụng. Ví dụ, một mô hình phân loại cảm xúc bài đánh giá phim nên được sử dụng trên các bài đánh giá IMDB mới, chứ không phải trên các bài đánh giá nhà hàng trên Yelp hay các bài cập nhật trên Twitter. Nếu bạn muốn đánh giá cảm xúc của một tweet, hãy bắt đầu bằng cách thu thập và gắn nhãn các tweet thực tế từ một nhóm người dùng tương tự với những người bạn dự đoán sẽ có trong môi trường sản xuất. Nếu không thể huấn luyện mô hình trên dữ liệu sản xuất, hãy chắc chắn rằng bạn hiểu rõ sự khác biệt giữa dữ liệu huấn luyện và dữ liệu sản xuất, và bạn đang tích cực điều chỉnh để khắc phục những sự khác biệt này.

Một hiện tượng liên quan mà bạn cần chú ý là **không gian drift (concept drift)**. Bạn sẽ gặp phải hiện tượng này trong hầu hết các bài toán thực tế, đặc biệt là những bài toán liên quan đến dữ liệu do người dùng tạo ra. **Concept drift** xảy ra khi các đặc tính của dữ liệu sản xuất thay đổi theo thời gian, khiến độ chính xác của mô hình giảm dần. Ví dụ, một hệ thống gợi ý nhạc được huấn luyện vào năm 2013 có thể không còn hiệu quả vào ngày nay. Tương tự, bộ dữ liệu IMDB bạn đã làm việc có thể được thu thập vào năm 2011, và một mô hình huấn luyện trên nó có thể không hoạt động tốt với các bài đánh giá từ năm 2020 so với năm 2012, vì từ vựng, cách diễn đạt và thể loại phim đã thay đổi theo thời gian. **Concept drift** đặc biệt nghiêm trọng trong các tình huống đối kháng, như phát hiện gian lận thẻ tín dụng, nơi các mẫu gian lận thay đổi gần như mỗi ngày.

**Một Hiện Tượng Liên Quan Mà Bạn Cần Lưu Ý Là "Concept Drift"**

Bạn sẽ gặp phải **concept drift** trong hầu hết các bài toán thực tế, đặc biệt là những bài toán liên quan đến dữ liệu do người dùng tạo ra. **Concept drift** xảy ra khi các đặc tính của dữ liệu sản xuất thay đổi theo thời gian, dẫn đến sự suy giảm độ chính xác của mô hình. Ví dụ, một hệ thống gợi ý nhạc được huấn luyện vào năm 2013 có thể không còn hiệu quả trong ngày hôm nay. Tương tự, bộ dữ liệu IMDB mà bạn đã sử dụng có thể được thu thập vào năm 2011, và một mô hình huấn luyện trên bộ dữ liệu đó có thể không hoạt động tốt với các bài đánh giá từ năm 2020 so với năm 2012, vì từ vựng, cách diễn đạt và thể loại phim thay đổi theo thời gian. **Concept drift** đặc biệt nghiêm trọng trong các bối cảnh đối kháng, như phát hiện gian lận thẻ tín dụng, nơi các mẫu gian lận thay đổi gần như mỗi ngày. Để xử lý **concept drift** nhanh, bạn cần liên tục thu thập dữ liệu, gắn nhãn và huấn luyện lại mô hình.

Hãy nhớ rằng học máy chỉ có thể ghi nhớ các mẫu có trong dữ liệu huấn luyện của bạn. Bạn chỉ có thể nhận diện những gì bạn đã thấy trước đây. Việc sử dụng học máy huấn luyện trên dữ liệu quá khứ để dự đoán tương lai chính là giả định rằng tương lai sẽ hành xử giống như quá khứ. Tuy nhiên, điều này thường không đúng.

**Không Nên Xử Lý Dữ Liệu Như Một "Hộp Đen"**

Trước khi bắt đầu huấn luyện mô hình, bạn nên khám phá và trực quan hóa dữ liệu để tìm hiểu những yếu tố nào làm cho nó có khả năng dự đoán, từ đó giúp bạn trong việc kỹ thuật tạo đặc trưng (feature engineering) và phát hiện các vấn đề tiềm ẩn.

Dưới đây là một số điều cần làm khi khám phá dữ liệu của bạn:

* **Nếu dữ liệu của bạn chứa hình ảnh hoặc văn bản tự nhiên, hãy xem qua một vài mẫu (và nhãn của chúng) trực tiếp.** Điều này sẽ giúp bạn hiểu rõ hơn về dữ liệu và cách mà nhãn có thể liên quan đến các đặc trưng.
* **Nếu dữ liệu chứa các đặc trưng số, hãy vẽ đồ thị histogram của các giá trị đặc trưng** để nắm bắt phạm vi giá trị và tần suất của các giá trị khác nhau. Điều này có thể giúp bạn hiểu sự phân phối của các đặc trưng.
* **Nếu dữ liệu của bạn bao gồm thông tin về vị trí, hãy vẽ dữ liệu lên bản đồ.** Liệu có những mẫu rõ ràng nào xuất hiện không? Điều này có thể giúp bạn xác định các mối quan hệ không gian trong dữ liệu.
* **Liệu một số mẫu dữ liệu có thiếu giá trị cho một số đặc trưng không?** Nếu có, bạn sẽ cần xử lý vấn đề này trong quá trình chuẩn bị dữ liệu (chúng ta sẽ đề cập đến cách làm này ở phần tiếp theo).
* **Nếu bài toán của bạn là phân loại, hãy in số lượng mẫu của mỗi lớp trong dữ liệu.** Các lớp có được phân bổ đều không? Nếu không, bạn sẽ cần phải điều chỉnh để xử lý sự mất cân bằng giữa các lớp.
* **Kiểm tra sự rò rỉ mục tiêu (target leakage):** Đây là hiện tượng khi có các đặc trưng trong dữ liệu của bạn cung cấp thông tin về mục tiêu mà bạn đang cố gắng dự đoán, và những thông tin này có thể không có sẵn trong môi trường sản xuất. Ví dụ, nếu bạn đang huấn luyện mô hình trên hồ sơ y tế để dự đoán liệu ai đó có được điều trị ung thư trong tương lai, và hồ sơ bao gồm đặc trưng “người này đã được chẩn đoán mắc ung thư,” thì mục tiêu của bạn đang bị rò rỉ vào dữ liệu. Hãy luôn hỏi bản thân: Mỗi đặc trưng trong dữ liệu của bạn có phải là thứ sẽ có sẵn dưới dạng giống hệt trong môi trường sản xuất không?