TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

XÂY DỰNG HỆ THỐNG ĐỊNH GIÁ TÀI SẢN TỰ ĐỘNG ỨNG DỤNG HỌC MÁY

Sinh viên thực hiện : Quách Huy Tùng
Lớp CN ĐTVT 02 - K60

Giảng viên hướng dẫn : PGS. TS Phạm Văn Bình

Hà Nội, 6 - 2020

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG

ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

Đề tài:

XÂY DỰNG HỆ THỐNG ĐỊNH GIÁ TÀI SẢN TỰ ĐỘNG ỨNG DỤNG HỌC MÁY

Sinh viên thực hiện : Quách Huy Tùng
Lớp CN ĐTVT 02 - K60

Giảng viên hướng dẫn : PGS. TS Phạm Văn Bình

Cán bộ phản biện :

Hà Nội, 6 - 2020

ĐÁNH GIÁ QUYỂN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho giảng viên hướng dẫn)

Tên giảng viên hướng dẫn: . . . . . .
Họ và tên sinh viên: . . . . . MSSV: . . . . . .
Tên đồ án: . . . . . .
. . . . . .

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:
Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

Có sự kết hợp giữa lý thuyết và thực hành (20)						
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án	1	2	3	4	5
2	Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế)	1	2	3	4	5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề	1	2	3	4	5
4	Có kết quả mô phỏng/thực nghiệm và trình bày rõ ràng kết quả đạt được	1	2	3	4	5
Có khả năng phân tích và đánh giá kết quả (15)						
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống	1	2	3	4	5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng	1	2	3	4	5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai	1	2	3	4	5
Kỹ năng viết quyển đồ án (10)						
8	Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến; căn lề thống nhất, có dấu cách sau dấu chấm, dấu phẩy v.v.), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định	1	2	3	4	5
9	Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.)	1	2	3	4	5
Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)						
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/Đạt giải SVNCKH giải 3 cấp Viện trở lên/Có giải thưởng khoa học (quốc tế hoặc trong nước) từ giải 3 trở lên/Có đăng ký bằng phát minh, sáng chế	5				
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị SVNCKH nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành (VD: TI contest)	3				
10c	Không có thành tích về nghiên cứu khoa học	0				
Điểm tổng		**/50**				
Điểm tổng quy đổi về thang 10						

Nhận xét khác (về thái độ và tinh thần làm việc của sinh viên)

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Ngày: . . . / . . . / 20 . . .
Người nhận xét
(Ký và ghi rõ họ tên)

ĐÁNH GIÁ QUYỂN ĐỒ ÁN TỐT NGHIỆP

(Dùng cho cán bộ phản biện)

Tên giảng viên hướng dẫn: . . . . . .
Họ và tên sinh viên: . . . . . MSSV: . . . . . .
Tên đồ án: . . . . . .
. . . . . .

Chọn các mức điểm phù hợp cho sinh viên trình bày theo các tiêu chí dưới đây:
Rất kém (1); Kém (2); Đạt (3); Giỏi (4); Xuất sắc (5)

Có sự kết hợp giữa lý thuyết và thực hành (20)						
1	Nêu rõ tính cấp thiết và quan trọng của đề tài, các vấn đề và các giả thuyết (bao gồm mục đích và tính phù hợp) cũng như phạm vi ứng dụng của đồ án	1	2	3	4	5
2	Cập nhật kết quả nghiên cứu gần đây nhất (trong nước/quốc tế)	1	2	3	4	5
3	Nêu rõ và chi tiết phương pháp nghiên cứu/giải quyết vấn đề	1	2	3	4	5
4	Có kết quả mô phỏng/thực nghiệm và trình bày rõ ràng kết quả đạt được	1	2	3	4	5
Có khả năng phân tích và đánh giá kết quả (15)						
5	Kế hoạch làm việc rõ ràng bao gồm mục tiêu và phương pháp thực hiện dựa trên kết quả nghiên cứu lý thuyết một cách có hệ thống	1	2	3	4	5
6	Kết quả được trình bày một cách logic và dễ hiểu, tất cả kết quả đều được phân tích và đánh giá thỏa đáng	1	2	3	4	5
7	Trong phần kết luận, tác giả chỉ rõ sự khác biệt (nếu có) giữa kết quả đạt được và mục tiêu ban đầu đề ra đồng thời cung cấp lập luận để đề xuất hướng giải quyết có thể thực hiện trong tương lai	1	2	3	4	5
Kỹ năng viết quyển đồ án (10)						
8	Đồ án trình bày đúng mẫu quy định với cấu trúc các chương logic và đẹp mắt (bảng biểu, hình ảnh rõ ràng, có tiêu đề, được đánh số thứ tự và được giải thích hay đề cập đến; căn lề thống nhất, có dấu cách sau dấu chấm, dấu phẩy v.v.), có mở đầu chương và kết luận chương, có liệt kê tài liệu tham khảo và có trích dẫn đúng quy định	1	2	3	4	5
9	Kỹ năng viết xuất sắc (cấu trúc câu chuẩn, văn phong khoa học, lập luận logic và có cơ sở, từ vựng sử dụng phù hợp v.v.)	1	2	3	4	5
Thành tựu nghiên cứu khoa học (5) (chọn 1 trong 3 trường hợp)						
10a	Có bài báo khoa học được đăng hoặc chấp nhận đăng/Đạt giải SVNCKH giải 3 cấp Viện trở lên/Có giải thưởng khoa học (quốc tế hoặc trong nước) từ giải 3 trở lên/Có đăng ký bằng phát minh, sáng chế	5				
10b	Được báo cáo tại hội đồng cấp Viện trong hội nghị SVNCKH nhưng không đạt giải từ giải 3 trở lên/Đạt giải khuyến khích trong các kỳ thi quốc gia và quốc tế khác về chuyên ngành (VD: TI contest)	3				
10c	Không có thành tích về nghiên cứu khoa học	0				
Điểm tổng		**/50**				
Điểm tổng quy đổi về thang 10						

Nhận xét khác của cán bộ phản biện

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Ngày: . . . / . . . / 20 . . .
Người nhận xét
(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Cuộc cách mạng khoa học kỹ thuật, Chuyển đổi số đã thay đổi bộ mặt cuộc sống của chúng ta. Từ sự phát minh các hệ thống thông tin, liên lạc, mạng Internet, các ứng dụng, dịch vụ đã thay đổi tận gốc rễ nhiều phương diện đời sống xã hội. Cách mạng công nghiệp lần thứ 4, hay còn được biết đến với tên gọi Chuyển đổi số, với nền móng là Trí tuệ nhân tạo, Dữ liệu lớn, công nghệ Blockchain, ... để tối ưu hóa qui trình vận hành, sản xuất, quảng cáo, bán hàng. Nó cũng thay đổi tận cốt lõi nhiều ngành truyền thống như sự ra đời của các nền tảng tài chính ngang hàng, ngân hàng điện tử, mang đến cơ hội tiếp cận vốn dễ dàng hơn.

Đồ án tốt nghiệp của em hướng đến ứng dụng công nghệ Học máy - một nhánh của Trí tuệ nhân tạo, trong bài toán định giá tài sản cho doanh nghiệp tài chính. Việc tích hợp công cụ định giá tự động sẽ giúp cho doanh nghiệp tài chính có thể nhanh chóng cung cấp cho khách hàng các dịch vụ của mình. Qua đây, em cũng xin gửi lời cám ơn đến PGS.TS Phạm Văn Bình đã quan tâm và hướng dẫn tận tình, đồng thời giúp em có những định hướng và mục tiêu rõ ràng cho bản thân sau khi tốt nghiệp.

LỜI CAM ĐOAN

Tôi là Quách Huy Tùng, mã số sinh viên 20156822, sinh viên lớp Cử nhân điện tử 02, khóa K60. Người hướng dẫn là PGS.TS. Phạm Văn Bình. Tôi xin cam đoan toàn bộ nội dung được trình bày trong đồ án Xây dựng hệ thống định giá tài sản tự động ứng dụng mô hình học máy là kết quả quá trình tìm hiểu và triển khai của tôi. Các dữ liệu được sử dụng trong đồ án hoàn toàn là thực tế và không vi phạm bản quyền. Mọi thông tin trích dẫn đều tuân thủ các quy định về sở hữu trí tuệ; các tài liệu thao khảo được liệt kê rõ ràng. Tôi xin chịu hoàn toàn trách nhiệm với những nội dung được viết trong đồ án này.

Hà Nội, ngày . . . , tháng . . . , năm 20 . . .

Người cam đoan

Mục lục

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

TÓM TẮT ĐỒ ÁN

Trong những năm gần đây, việc ứng dụng công nghệ vào lĩnh vực tài chính ngân hàng là một xu hướng phát triển mạnh. Các công ty Fin-tech ra đời, nhằm mục tiêu đưa nguồn vốn dễ dàng đến tay người tiêu dùng. Các công ty này chủ yếu hoạt động theo mô hình cho vay ngang hàng, cho vay trả góp hoặc cầm cố tài sản.

Một trong các nghiệp vụ quan trọng của các doanh nghiệp này là việc định giá tài sản thế chấp của khách hàng. Việc này đòi hỏi các doanh nghiệp này phải tiêu tốn chi phí cho nhân sự các phòng ban, chuyên viên định giá tài sản. Với mô hình cho vay qua ứng dụng, trang web, các yêu cầu lập hợp đồng cho vay của các ứng dụng này là rất lớn, điều đó tạo gánh nặng lên các chuyên viên định giá tài sản.

Từ thực trạng đó, trong đồ án này em trình bày một cách tiếp cận mới cho việc định giá tài sản tự động. Mô hình định giá tự động được xây dựng dựa trên kỹ thuật Học máy, với kỳ vọng sẽ có thể tự động hóa công việc định giá tài sản và nâng cao hiệu suất làm việc cho doanh nghiệp tài chính cũng như trải nghiệm cho người sử dụng dịch vụ.

Chương 1

BÀI TOÁN ĐỊNH GIÁ TÀI SẢN TỰ ĐỘNG CHO DOANH NGHIỆP TÀI CHÍNH

1.1 Đặt vấn đề

Trong giai đoạn hiện nay, các ứng dụng tài chính cá nhân nở rộ xu hướng cho vay ngang hàng P2P, các dịch vụ cầm đồ, ngân hàng điện tử. Ở Việt Nam, các doanh nghiệp tài chính công nghệ(Fintech) đặc biệt phát triển và mở rộng trong giai đoạn 10 năm trở lại đây như F88, Tima, ... đã trở thành nguồn cung cấp vốn nhanh chóng, tiện lợi cho các cá nhân có nhu cầu. Để đảm bảo tính tin cậy, các doanh nghiệp này thường phải yêu cầu người dùng thế chấp tài sản, hoặc giấy tờ. Sau đó các doanh nghiệp này có bộ phận định giá tài sản để lấy căn cứ nhằm hạn mức mức cho vay.

Việc định giá tài sản này yêu cầu một bộ phận của các doanh nghiệp này phải có kinh nghiệm trong lĩnh vực này. Do cần có một lượng nhân sự đủ để phù hợp với lượng yêu cầu gia tăng từ phía khách hàng, các doanh nghiệp phải duy trì một lượng chuyên gia thẩm định tài sản. Điều đó gây tốn kém cho các doanh nghiệp này. Hơn nữa, các sai sót của các cá nhân là khó có thể tránh khỏi, gây vấn đề thất thoát tài sản cho doanh nghiệp.

1.2 Mục tiêu và giới hạn của đồ án

Từ yêu cầu trên, dưới sự hướng dẫn của PGS.TS Phạm Văn Bình, em xây dựng đồ án với mục tiêu giải quyết hiệu quả bài toán định giá tài sản tự động cho các doanh nghiệp tài chính, ngân hàng trong thời đại Chuyển đổi số hiện nay.

Do lĩnh vực này là một lĩnh vực rộng, có nhiều loại tài sản cần được định giá với những kiến thức chuyên môn và phương pháp khác nhau. Chủ yếu với các ngân hàng thì yêu cầu chủ yếu là xây dựng được hệ thống định giá tài sản cố định như nhà đất hoặc ô tô. Trong khi với các công ty cho vay ngang hàng, cầm đồ thì lại cần đến các hệ thống định giá cho các tài sản có giá trị thấp hơn như xe máy, điện thoại, máy tính, ... Như vậy, em lựa chọn một lĩnh vực nhỏ là định giá tài sản ô tô cũ, phục vụ cho các hệ thống thẩm định giá trị tài sản của ngân hàng.

Với những yêu cầu của đồ án, em đặt mục tiêu đồ án này xây dựng được một hệ thống:

- Hệ thống được xây dựng là một dịch vụ con trong hệ thống đa dịch vụ(Microservices) của doanh nghiệp, hoạt động theo mô hình JSON API.
- Hệ thống có khả năng định giá ô tô cũ thuộc nhiều hãng, dòng xe, nhiều loại xe.
- Hệ thống có khả năng định giá với độ chính xác cao. Giá trị dự đoán không sai khác quá 5% với giá trị của xe được định giá bởi chuyên viên.
- Hệ thống hoạt động ổn định, với độ trễ thấp hơn 20 ms với mỗi yêu cầu dự đoán.

1.3 Phương pháp thực hiện đồ án

Trong đồ án này, em sử dụng các mô hình học máy để xây dựng hệ thống. Học máy là một công cụ mạnh mẽ để xây dựng các mô hình dự đoán, phân loại cũng như gom nhóm dữ liệu. Học máy sử dụng một lượng lớn dữ liệu để học được "quy luật ẩn" trong dữ liệu, sau đó sử dụng quy luật đã học được để đưa ra suy luận trên dữ liệu mới.

Với sự phát triển của Internet tại nước ta trong suốt 20 năm qua, các diễn đàn, trang web thương mại điện tử phát triển mạnh mẽ, kinh doanh đa dạng ngành nghề. Các diễn đàn, sàn giao dịch ô tô cũ như otofun.vn, bonbanh.com thu hút được một lượng người dùng lớn, kéo theo đó là một lượng dữ liệu tương đối về hoạt động rao bán, kinh doanh xe cũ. Đây là nguồn dữ liệu tốt cho việc khai thác, sử dụng cho các thuật toán khai phá dữ liệu để xây dựng mô hình định giá tự động.

Trong đồ án này, em sẽ sử dụng dữ liệu được thu thập từ các website, nền tảng rao bán xe cũ để làm dữ liệu huấn luyện, kiểm thử để lựa chọn mô hình học máy phù hợp với yêu cầu của đồ án. Mô hình học máy sẽ được triển khai trên nền tảng JSON API.

1.4 Phân chia các gói công việc và kế hoạch thực hiện

Để thực hiện đồ án này, em phân chia đồ án thành các gói công việc sau:

Số thứ tự	Tên gói công việc	Khoảng thời gian dự kiến	Ngày bắt đầu	Ngày kết thúc
1	Tìm hiểu lý thuyết các mô hình học máy, các kiến thức về thu thập, xử lý dữ liệu, xây dựng dịch vụ JSON Web API	1 tuần	24/02/2020	02/03/2020
2	Xây dựng trình thu thập dữ liệu	2 tuần	03/03/2020	17/03/2020
3	Xử lý dữ liệu sau khi thu thập, trực quan hóa và loại nhiễu	1 tuần	18/02/2020	25/03/2020
4	Xây dựng mô hình học máy	6 tuần	26/03/2020	10/05/2020
5	Triển khai sản phẩm	1 tuần	11/05/2020	19/05/2020
6	Đánh giá, kiểm thử hệ thống	1 tuần	19/05/2020	26/05/2020
7	Viết báo cáo, tổng kết kết quả của đồ án	4 tuần	19/05/2020	19/06/2020
	Tổng kết	4 tháng	24/02/2020	20/06/2020

Bảng 1.1: Phân chia các gói công việc và thời gian thực hiện

Chương 2

NỀN TẢNG LÝ THUYẾT VÀ CÁC NỀN TẢNG CÔNG NGHỆ

Để thực hiện được các gói công việc đề ra, qua sự định hướng của thầy, em tìm hiểu các nền tảng lý thuyết và công nghệ. Cụ thể, trong chương này, em tìm hiểu về cách thức hoạt động của các ứng dụng Web, cách xây dựng trình thu thập dữ liệu Web; lý thuyết Học Máy và các giải thuật Học máy được áp dụng. Phần cuối chương trình bày về các công cụ hỗ trợ việc cài đặt trình thu thập, xử lý và trực quan hóa dữ liệu, cũng như các công cụ cài đặt, kiểm thử và triển khai mô hình Học Máy.

2.1 Internet và ứng dụng Web

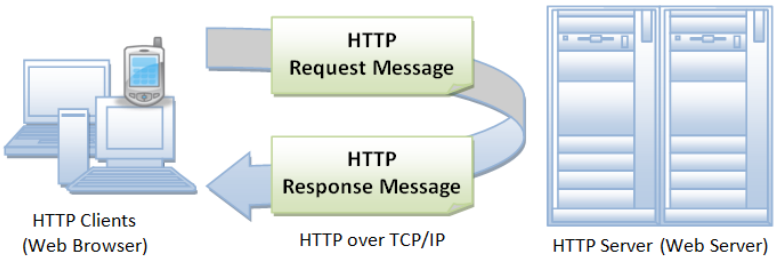
2.1.1 Giao thức HTTP và ứng dụng Web

Giao thức HTTP

HTTP là giao thức cấp độ ứng dụng cho các hệ thống thông tin phân phối, cộng tác, đa phương tiện, là nền tảng cho giao tiếp dữ liệu cho WWW (ví dụ: Internet) từ 1990. HTTP là một giao thức chung và không trạng thái (stateless) mà có thể được sử dụng cho các mục đích khác cũng như các sự co giãn của các phương thức yêu cầu, các code lỗi và Header của nó.

Về cơ bản, HTTP là một giao thức giao tiếp trên cơ sở TCP/IP, mà được sử dụng để phân phối dữ liệu (các tệp HTML, các file ảnh, ...) trên WWW. Cổng mặc định là TCP 80, những các cổng khác cũng có thể được sử dụng. Nó cung cấp một cách được tiêu chuẩn hóa cho các máy tính để giao tiếp với nhau. Chi tiết kỹ thuật HTTP xác định cách mà dữ liệu yêu cầu của Client sẽ được xây dựng và được gửi tới Server, và cách để Server phản hồi các yêu cầu này.

Sơ đồ hoạt động của HTTP



Hình 2.1: Giao thức HTTP

HTTP hoạt động dựa trên mô hình Client – Server. Trong mô hình này, các máy tính của người dùng sẽ đóng vai trò làm máy khách (Client). Sau một thao tác nào đó của người dùng, các máy khách sẽ gửi yêu cầu đến máy chủ (Server) và chờ đợi câu trả lời từ những máy chủ này.
HTTP cho phép tạo các yêu cầu gửi và nhận các kiểu dữ liệu, do đó cho phép xây dựng hệ thống độc lập với dữ liệu được truyền giao.

Uniform Resource Locator(URL)

Một URL (Uniform Resource Locator) được sử dụng để xác định duy nhất một tài nguyên trên Web. Một URL có cấu trúc như sau: **protocol://hostname:port/path-and-file-name**.
Trong một URL có 4 thành phần:

- **protocol**: giao thức tầng ứng dụng được sử dụng bởi client và server
- **hostname**: tên DNS domain
- **port**: Cổng TCP để server lắng nghe request từ client
- **path-and-file-name**: Tên và vị trí của tài nguyên yêu cầu.

HTTP Method

HTTP Method	Hành động
GET	Truy cập, xem một tài nguyên xác định trên máy chủ
POST	Yêu cầu máy chủ chấp nhận thực thể được đính kèm trong request, ví dụ như thông tin hoặc file tải lên
PUT	Nếu tài nguyên đã có, nó sẽ bị sửa đổi theo thông tin mới được tải lên. Nếu chưa có, sẽ tạo ra một tài nguyên theo thông tin được tải lên.
PATCH	Áp dụng cho việc sửa đổi một phần của tài nguyên
DELETE	Xóa tài nguyên

Bảng 2.1: Các phương thức HTTP

HTTP Status Code

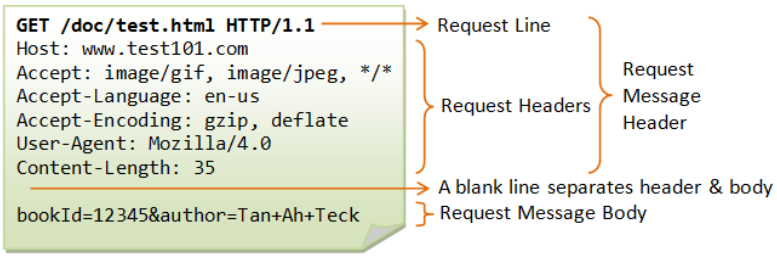
Thông báo về trạng thái kết quả khi nhận được yêu cầu và xử lí bên server cho client.

HTTP Status Code	Trạng thái	Ví dụ
1xx	Thông tin	100: Continue
2xx	Thành công	201: Created
3xx	Chuyển hướng	302: Found after redirect
4xx	Lỗi client	404: Resources not found
5xx	Lỗi server	500: Internal server error

Bảng 2.2: HTTP Status Code

Các thành phần chính của HTTP

HTTP Requests



Hình 2.2: Cấu trúc HTTP Request

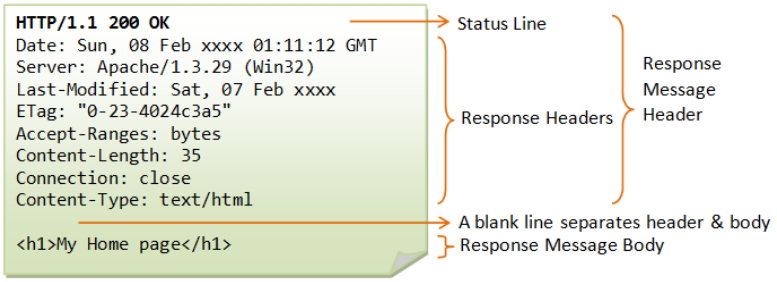
Cấu trúc của một HTTP Request:

- Một Request-line = HTTP Method + URI–Request + Phiên bản HTTP. Giao thức HTTP định nghĩa một tập các giao thức GET, POST, HEAD, PUT... Client có thể sử dụng một trong các phương thức đó để gửi request lên server.
- Các trường header: Các trường header cho phép client truyền thông tin bổ sung về yêu cầu, và về chính client, đến server. Một số trường: Accept-Charset, Accept-Encoding, Accept-Language, Authorization, Expect, From, Host, ...
- Một dòng trống để đánh dấu sự kết thúc của các trường Header.
- Tùy chọn một thông điệp

Khi request đến server, server thực hiện một trong 3 hành động sau:

- Server phân tích request nhận được, maps yêu cầu với tập tin trong tập tài liệu của server, và trả lại tập tin yêu cầu cho client.
- Server phân tích request nhận được, maps yêu cầu vào một chương trình trên server, thực thi chương trình và trả lại kết quả của chương trình đó.
- Request từ client không thể đáp ứng, server trả lại thông báo lỗi.

HTTP Responses



Hình 2.3: Cấu trúc HTTP Response

Cấu trúc của một HTTP response:

- Một Status-line = Phiên bản HTTP + Mã trạng thái + Trạng thái
- Có thể có hoặc không có các trường header
- Một dòng trống để đánh dấu sự kết thúc của các trường header
- Tùy chọn một thông điệp

Các đặc trưng cơ bản của HTTP

HTTP là một giao thức nhỏ gọn, hướng đến sự tinh gọn và tiêu chuẩn hóa cách truy cập các tài nguyên Web. Nhưng nhờ 3 đặc điểm sau, tuy nhỏ gọn nhưng HTTP là một giao thức mạnh mẽ:

- **HTTP là giao thức connectionless (kết nối không liên tục)**: Client của HTTP, ví dụ: một trình duyệt khởi tạo một yêu cầu HTTP và sau đó một yêu cầu được tạo ra, Client ngắt kết nối từ Server và đợi cho một phản hồi. Server xử lý yêu cầu và thiết lập lại sự kết nối với Client để gửi phản hồi trở lại.
- **HTTP là một phương tiện độc lập**: Nó nghĩa là, bất kỳ loại dữ liệu nào cũng có thể được gửi bởi HTTP miễn là Server và Client biết cách để kiểm soát nội dung dữ liệu. Nó được yêu cầu cho Client cũng như Server để xác định kiểu nội dung bởi sử dụng kiểu MIME thích hợp.
- **HTTP là stateless**: Như đã được đề cập ở trên, HTTP là connectionless và nó một kết quả trực tiếp là HTTP trở thành một giao thức Stateless. Server và Client biết về nhau chi trong một yêu cầu hiện tại. Sau đó, cả hai chúng nó quên tất cả về nhau. Do bản chất của giao thức, cả Client và các trình duyệt có thể giữ lại thông tin giữa các yêu cầu khác nhau giữa các trang web.

Ứng dụng web

Ứng dụng Web là ứng dụng được triển khai trên nền tảng Internet, theo mô hình Client/Server, giao tiếp giữa chúng thông qua giao thức HTTP. Tích hợp dữ liệu giữa Client/Server dựa trên các loại dữ liệu bán cấu trúc như HTML, JSON, XML, ...

2.1.2 Ngôn ngữ đánh dấu siêu văn bản HTML và mô hình DOM

Các website được trình bày bằng ngôn ngữ đánh dấu HTML. Được coi như xương sống của bất cứ website nào, HTML chứa các thông tin được hiển thị trên trang web, cũng như các thông tin ẩn như các thẻ meta, ... HTML được biểu diễn dưới dạng cây, gọi là mô hình DOM.

HTML

HTML là chữ viết tắt của cụm từ HyperText Markup Language) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML. Cha đẻ của HTML là Tim Berners-Lee, cũng là người khai sinh ra World Wide Web và chủ tịch của World Wide Web Consortium (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

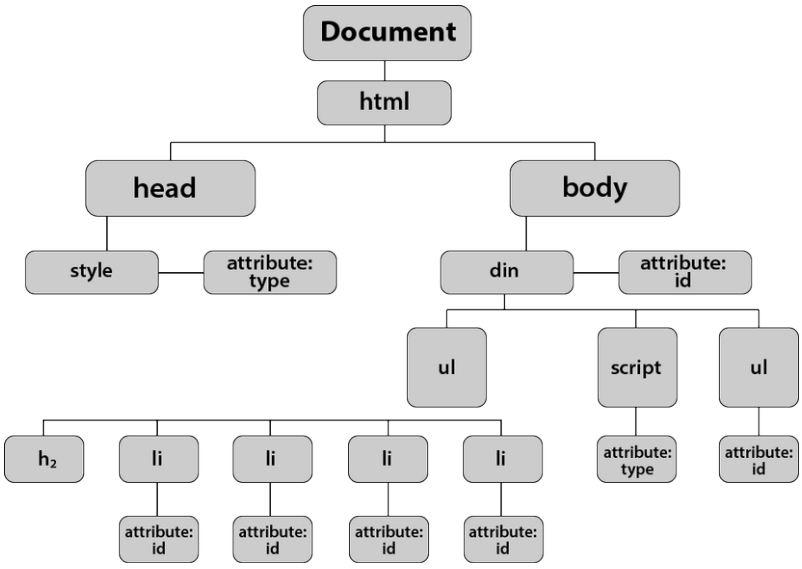
Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc ngọn (ví dụ <html>) và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví dụ

 và). Các văn bản muốn được đánh dấu bằng HTML sẽ được khai báo bên trong cặp thẻ (ví dụ Đây là chữ in đậm). Nhưng một số thẻ đặc biệt lại không có thẻ đóng và dữ liệu được khai báo sẽ nằm trong các thuộc tính (ví dụ như thẻ <img>).

Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm.

DOM

Các trang HTML được biểu diễn thành dạng cây, gọi là DOM. Một DOM đơn giản có cấu trúc như sau:



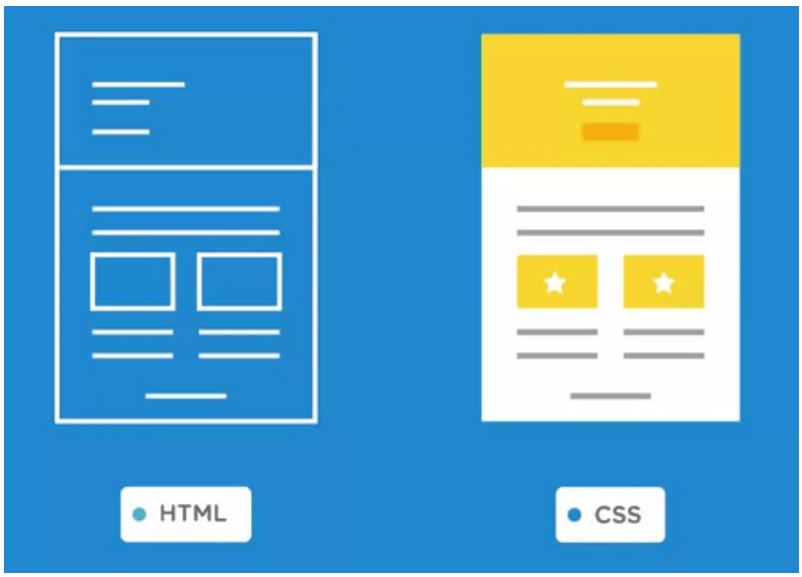
Hình 2.4: Mô hình DOM

DOM tạo ra một giao diện lập trình dễ dàng để lập trình viên tương tác với webpage. Thông qua việt duyệt cây, truy vấn nút trên cây, thay đổi dữ liệu trên nút, lập trình viên có thể tạo ra các website giàu tương tác, đẹp mắt.

2.1.3 CSS & Javascript

CSS

CSS là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,... thì CSS sẽ giúp chúng ta có thể thêm một chút "phong cách" vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc,...rất nhiều.



Hình 2.5: Minh họa CSS

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Cấu trúc một đoạn CSS

Một đoạn CSS có cấu trúc gồm 3 thành phần sau:

- Vùng chọn
- Thuộc tính
- Giá trị

Nghĩa là nó sẽ được khai báo bằng vùng chọn, sau đó các thuộc tính và giá trị sẽ nằm bên trong cặp dấu ngoặc nhọn . Mỗi thuộc tính sẽ luôn có một giá trị riêng, giá trị có thể là dạng số, hoặc các tên giá trị trong danh sách có sẵn của CSS. Phần giá trị và thuộc tính phải được cách nhau bằng dấu hai chấm, và mỗi một dòng khai báo thuộc tính sẽ luôn có dấu chấm phẩy ở cuối. Một vùng chọn có thể sử dụng không giới hạn thuộc tính.

Javascript

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). Khi nằm bên trong một môi trường (host environment), JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng (object).
JavaScript chứa các thư viện tiêu chuẩn cho các object, ví dụ như: Array, Date, và Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

- **Client-side JavaScript**: JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó.

Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.

- **Server-side JavaScript**: JavaScript phía Server, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

2.1.4 Truy vấn DOM

DOM cung cấp giao diện lập trình để lập trình viên có thể dễ dàng thao tác, chỉnh sửa nội dung trang web, cũng như truy cập các thông tin. Về cơ bản DOM cung cấp 2 cách thức truy vấn các phần tử là CSS Selector và XPATH Selector.

CSS Selector

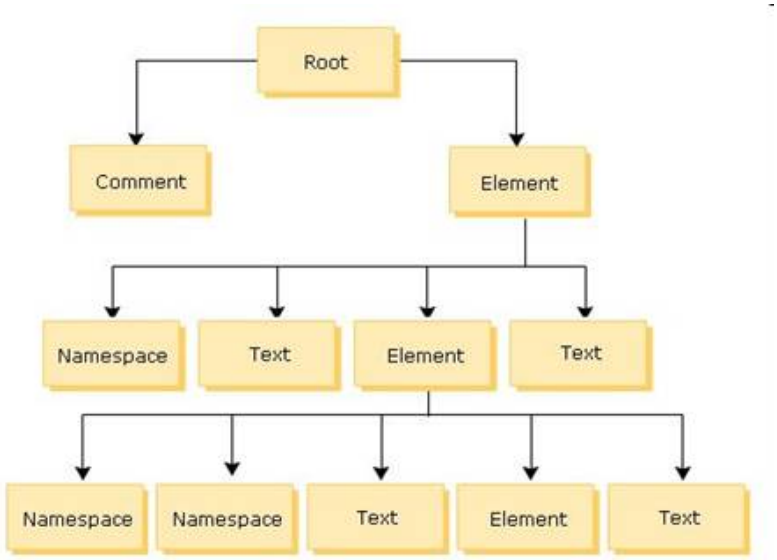
CSS Selector dùng để chọn, truy vấn các phần tử HTML. Trong một trang HTML, có rất nhiều thẻ giống nhau, nên ta thường đặt ID, Class để phân biệt. CSS Selector thường sử dụng ID, Class, tên thẻ, cấu trúc của trang HTML để chọn phần tử.
CSS Selector chia làm 3 loại:

- CSS Selector phân cấp. Ví dụ: div p strong
- CSS Selector ID. Ví dụ: div#active
- CSS Selector Class. Ví dụ: div.bg-yellow

XPATH Selector

XPATH là một thành phần hỗ trợ giúp truy xuất thông tin trong tập tin XML làm tiền đề cho việc áp dụng stylesheet kết hợp XML để tạo ra kết xuất tùy theo yêu cầu. Bên cạnh đó XPath cũng làm nền tảng cho việc hỗ trợ truy vấn parsing dữ liệu của tài liệu XML cực kỳ nhanh chóng hiệu quả. Hơn thế nữa, XPath hỗ trợ nền tảng để tạo ra XQuery áp dụng trong truy vấn dữ liệu tương tự như truy vấn SQL trên cơ sở dữ liệu. Thông qua việc sử dụng biểu thức XPath để định hướng tìm kiếm dữ liệu trên XML thay vì phải thực hiện tìm kiếm đệ qui để duyệt cây XML. XPATH còn được mở rộng, hỗ trợ cả truy vấn HTML.

Xpath định nghĩa 7 loại nodes theo mô hình thể hiện bên dưới từ root, element, attribute, text, namespace, processing-instruction và comment.



Hình 2.6: Cấu trúc XPATH

Ngoài ra, Xpath còn định nghĩa một số node đặc biệt để thể hiện mối quan hệ giữa các node trong mô hình trong quá trình xử lý như sau:

- Parent Node: node trên trực tiếp của node hiện hành
- Child Node: tập node trực tiếp của node hiện hành cấp thấp hơn
- Sibling: node ngang hàng hay cùng cha với node hiện hành
- Ancestors: tất cả node con bên trên node hiện hành cùng nhánh
- Descendants: tất cả node con bên dưới của node hiện hành cùng nhánh

Xpath Data Model được định nghĩa là duyệt toàn bộ cây nội dung XML và chuyển đổi – mapping chúng thành 7 loại node đã mô tả ở trên và XPath thực hiện truy vấn trên nội dung cây XML đã được tạo ra. Mỗi node trong XPath đều có giá trị kiểu chuỗi chứa thông tin của một node bao gồm localname và namespace nếu có để có thể truy vấn đến node trong mô hình một cách dễ dàng thông qua tên.

Cú pháp của XPath:

- Để truy vấn với đường dẫn tuyệt đối nghĩa là đi từ root của tài liệu XML đến các thành phần cần truy cập, XPath qui định với cú pháp bắt đầu bằng dấu /
- Để truy vấn với đường dận tương đối để có thể truy cập đến thành phần bất kỳ thỏa điều kiện, XPath qui định cú pháp sử dụng với dấu //
- Để truy vấn đến một thành phần bất kỳ mà không cần biết tên của nó là gì, XPath qui định ký tự sử dụng là *
- Để truy cập thuộc tính của một node, XPath qui định thuộc tính truy vấn phải có cú pháp bắt đầu là @. Ví dụ @tênThuộcTính
- Để truy cập đến giá trị của một biến được định nghĩa, XPath qui định biến truy vấn phải có cú pháp bắt đầu là $. Ví dụ $tênBiến
- Điều kiện khi truy vấn được đặt trong dấu []

2.2 Thu thập dữ liệu Web

Trong thời đại chuyển đổi số, dữ liệu được ví như "mỏ vàng", đem lại lợi nhuận lớn cho doanh nghiệp. Từ dữ liệu về hành vi người dùng, dữ liệu về sản phẩm, thị trường, các doanh nghiệp có thể có cái nhìn sâu sắc về thị hiếu người dùng, từ đó có thể đưa ra chiến lược kinh doanh phù hợp. Ngoài ra, thu thập dữ liệu Web còn có ý nghĩa quan trọng trong các máy tìm kiếm như Google, Yahoo, Yandex, ...

2.2.1 Định nghĩa thu thập dữ liệu Web

Thu thập dữ liệu web, là một chương trình duyệt được lập trình để duyệt các website một cách tự động với mục đích trích rút, lưu trữ dữ liệu trên nền tảng Web.
Tùy vào mục đích của việc thu thập dữ liệu, ta có thể chia thu thập dữ liệu web ra thành hai loại: Web scraping và Web crawling.

2.2.2 Web scraping và Web crawling

Web Crawling là quá trình khai phá cấu trúc của Web, lưu trữ trang web để phục vụ cho các máy tìm kiếm. Các trình Web Crawling không nhằm mục đích trích chọn thông tin, mà thiên về việc lưu trữ, đánh chỉ mục tìm kiếm. Ngoài ra, các trình Web Crawling khai phá Web thường theo cấu trúc không xác định, bằng cách đi theo đường dẫn được nhúng trong trang Web hiện tại.

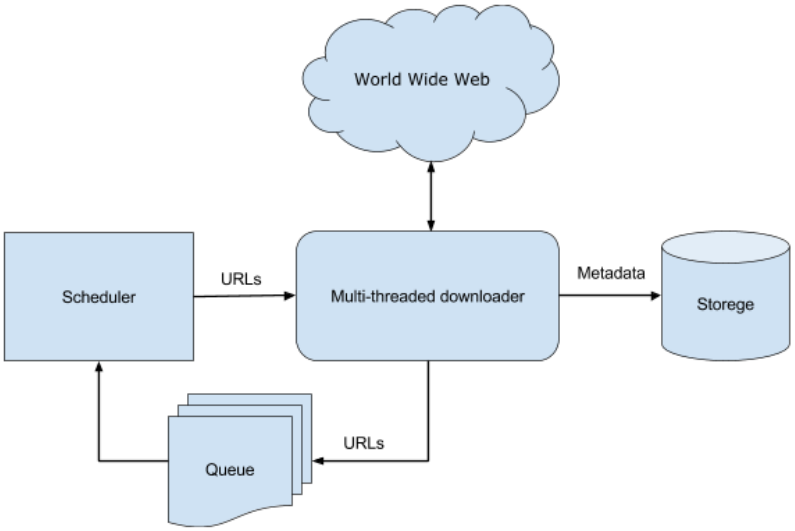
Ngược lại, Web Scraping lại là quá trình trích rút thông tin từ một hoặc một số trang web xác định, theo các tập luật xác định. Mục đích của Web Scraping là để thu thập, lưu trữ các thông tin cho việc lưu trữ dữ liệu hoặc phân tích dữ liệu.

	Web Crawling	Web Scraping
Mục đích	Lưu trữ trang web để tiến hành đánh chỉ mục, phục vụ tìm kiếm	Trích rút thông tin cần thiết để phân tích, sử dụng
Ứng dụng	Google, Bing, Yahoo	Baomoi, sosanhgia
Thông tin lưu trữ	Toàn bộ trang HTML	Chỉ các thông tin cần thiết
Cách điều hướng	Follow toàn bộ các link được trang trỏ tới	Điều hướng theo tập luật có sẵn

Bảng 2.3: So sánh Web Crawling và Web Scraping

Như vậy, với yêu cầu của đồ án là thu thập dữ liệu Web để phân tích, em lựa chọn mô hình Web Scraping

2.2.3 Lập trình trình thu thập dữ liệu Web tự động



Hình 2.7: Kiến trúc trình thu thập dữ liệu

Trình thu thập dữ liệu web thường có những thành phần sau:

- Hàng đợi các URL để xử lý
- Bộ lập lịch các yêu cầu HTTP
- Bộ download, bóc tách dữ liệu các trang web
- Bộ lưu trữ dữ liệu thu thập về

2.3 Học máy

2.3.1 Định nghĩa học máy

Theo Prof Mitchell: Machine Learning là 1 chương trình máy tính được nói là học hỏi từ kinh nghiệm E từ các tác vụ T và với độ đo hiệu suất P. Nếu hiệu suất của nó áp dụng trên tác vụ T và được đo lường bởi độ đo P tăng từ kinh nghiệm E.

Học máy đã trở thành một trong những chủ đề quan trọn nhất trong các tổ chức phát triển đang tìm kiếm cách thức đổi mới tận dụng tài sản dữ liệu để giúp doanh nghiệp đạt được một cấp độ mới sự hiểu biết. Với các mô hình học máy phù hợp, các tổ chứ có khả năng liên tục dự đoán những thay đổi trong kinh doanh để học có khả năng tốt nhất để dự đoán những gì tiếp theo. Khi dữ liệu liên tục được thêm vào, các mô hình học má đảm bảo giải pháp được cập nhật liên tục. Giá trị rất đơn giản: Nếu bạn sử dụng nguồn dữ liệu phù hợp nhất và liên tục thay đổi trong bối cảnh học máy, bạn có cơ hội dự đoán tương lại.
Học máy là một dạng của trí tuệ nhân tạo cho phép học từ dữ liệu hơn là thông qua lập trình rõ ràng. Tuy nhiên, học máy không phải là một quá trình đơn giản.

Học máy sử dụng nhiều thuật toán lặp đi lặp lại học hỏi từ dữ liệu để cải thiện, mô tả và dự đoán kết quả. Các thuật toán 'nuốt' dữ liệu học, sau đó có thể tạo ra các mô hình chính xác hơn dựa trên dữ liệu đó. Một mô hình học máy là một đầu ra được tạp ra khi bạn huấn luyện máy

của bạn học thuật toán với dữ liệu. Sau khi huấn luyện, khi bạn cung cấp một mô hình với một đầu vào, bạn sẽ nhận một output. Ví dụ, một thuật toán dự đoán sẽ tạo ra một mô hình dự đoán. Sau đó, khi bạn cung cấp mô hình dự đoán với dữ liệu, bạn sẽ nhận được một dự đoán dựa trên đữ liệu mà bạn đã huấn luyện mô hình. Học máy đang dần trở nên phổ biến cho việc tạo ra các mô hình phân tích.

2.3.2 Các bài toán học máy thông dụng

Bài toán phân loại:

Classification hay còn gọi là phân loại, phân lớp. Đây là một trong những bài toán được nghiên cứu nhiều nhất trong machine learning. Nhãn này thường là một phần tử trong một tập hợp có phần tử khác nhau. Mỗi phần tử trong tập hợp này được gọi là một lớp (class), và thường được đánh số từ 1 đến C. Để giải bài toán này, ta thường phải xây dựng một hàm số

$$f : R^d -> [1,2,3,...,C]$$

Khi đưa đầu vào x, mô hình sẽ tính $y = f(x)$, mô hình gán cho một điểm dữ liệu được mô tả bởi vector đặc trưng x một nhãn xác định bởi số y. Có một biến thể ở đầu ra của hàm số $f(x)$, khi đầu ra không phải là một số mà là một vector $y \in R^C$ trong đó chỉ ra xác suất để điểm dữ liệu rơi vào lớp c. Lớp được chọn cuối cùng là lớp có xác suất rơi vào là cao nhất. Việc sử dụng xác suất này đôi khi rất quan trọng, nó giúp chỉ ra độ chắc chắn (confidence) của mô hình. Nếu xác suất cao nhất là cao hơn nhiều so với các xác suất còn lại, ta nói mô hình có độ chắc chắn là cao khi phân lớp điểm dữ liệu . Ngược lại, nếu độ chênh lệch giữa xác suất cao nhất và các xác suất tiếp theo là nhỏ, thì khả năng mô hình đã phân loại nhầm là cao hơn.

Bài toán hồi quy (Regression)

Nếu nhãn không được chưa thành các nhóm mà là giá trị thực thì bài toán được gọi là hồi quy (regression). Trong bài toán này ta cần xây dựng hàm

$$f : R^d -> R$$

Bài toán regression có thể mở rộng ra việc dự đoán nhiều đầu ra cùng một lúc, khi đó, hàm cần tìm sẽ là $f : R^d -> R^m$ Một ví dụ là bài toán single image super resolution, ở đó, hệ thống cần đào tạo một bức ảnh có độ phân giải cao dựa trên một ảnh có độ phân giải thấp hơn. Khi đó, việc dự đoán giá trị của các pixel trong ảnh đầu ra là một bài toán regression với nhiều đầu ra.

Bài toàn phân cụm (Clustering)

Clustering là bài toán phân nhóm toàn bộ dữ liệu thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm. Ví dụ: Phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.

2.3.3 Đánh giá mô hình học máy

Phương pháp đánh giá

Chia tập dữ liệu D thành hai tập con không giao nhau.

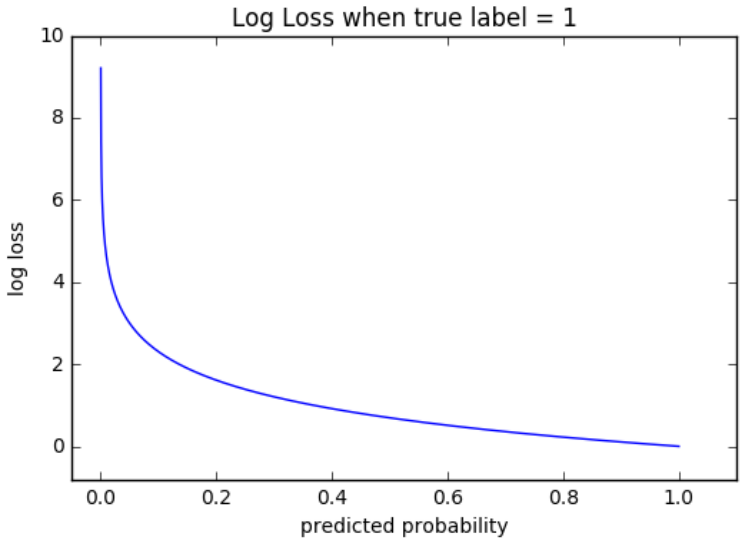
- Tập huấn luyện D_{train} để huấn luyện hệ thống.
- Tập huấn luyện D_{test} để kiểm thử hệ thống.
- $D = D_{\text{train}} + D_{\text{train}}$
- Bất kỳ ví dụ nào thuộc vào tập D_{test} đều không được sử dụng trong quá trình huấn luyện hệ thống.
- Bất kỳ ví dụ nào thuộc tập D_{train} đều không được sử dụng trong quá trình đánh giá hệ thống.
- Các ví dụ kiểm thử trong D_{test} cho phép một đánh giá không thiên vụ đối với hiệu năng hệ thống.

Các lựa chọn thường gặp: $|D_{\text{train}}| = 0.8|D|$, $|D_{\text{test}}| = 0.2|D|$.

Tiêu chí đánh giá mô hình phân loại

Hàm mất mát Cross-entropy
Hàm mất mát cross-entropy, hay log-loss, đánh giá hiệu quả của mô hình phân loại. Hàm mất mát Cross-entropy tăng khi giá trị dự đoán sai lệch càng lớn so với nhãn thật.

$$L = \sum_{i=1}^{n} y_i \cdot \log_2 p_{\text{o, i}}$$



Hình 2.8: Hàm Cross-entropy

Trong đó: $p_{\text{o, i}}$ đánh giá độ sai lệch của giá trị dự đoán của mô hình o so với giá trị nhãn tại ví dụ i.

Độ chính xác(Acc)

$$Acc = \frac{\text{Số phán đoán chính xác}}{\text{Tổng số phán đoán}}$$

Ma trận nhầm lẫn (confusion matrix)

- TP_i (true positive): Số lượng các ví dụ thuộc lớp c_i được phân loại chính xác vào lớp c_i.
- FP_i (false positive): Số lượng các ví dụ không thuộc lớp c_i bị phân loại nhầm vào lớp c_i.
- TN_i (true negative): Số lượng các ví dụ không thuộc lớp c_i được phân loại không vào lớp c_i.
- FN_i (true negative): Số lượng các ví dụ không thuộc lớp c_i được phân loại không vào lớp c_i.

Precision bằng tổng số ví dụ thuộc lớp c_i được phân loại chính xác chia cho tổng số các ví dụ được phân loại vào lớp c_i.

$$Precision(c_i) = \frac{TP_i}{TP_i + FN_i}$$

Recall bằng tổng số ví dụ thuộc lớp c_i được phân loại chính xác chia cho tổng số các ví dụ thuộc lớp c_i.

$$Recall(c_i) = \frac{TP_i}{TP_i + TN_i}$$

F1 score là một trung bình điều hòa (harmonic mean) của Precision và Recall. **Nhớ thêm phần nhận xét về hàm log loss**

$$F1 = \frac{Precision.Recall}{Precision + Recall}$$

Tiêu chí đánh giá mô hình hồi quy

L1-loss Hàm mất mát L1 đánh giá sai số tuyệt đối giữa giá trị dự đoán và giá trị thật sự của điểm dữ liệu theo công thức:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

L2-loss Hàm mất mát L2 đánh giá sai số bình phương giữa giá trị dự đoán và giá trị thật sự của điểm dữ liệu theo công thức:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

Như vậy, hàm mất mát L2 cho giá trị lớn hơn khi có nhiều điểm bất thường, ngoại lai trong bộ dữ liệu do hàm mất mát L2 được tính bằng cách bình phương sai số. Kết quả là hàm mất mát L1 có tính chống chịu lỗi tốt hơn và chịu ít ảnh hưởng hơn bởi các điểm dữ liệu bất thường. Trong khi đó, hàm mất mát L2 thường dẫn đến các mô hình rất "sát" với bộ dữ liệu, nhưng cũng nhạy cảm hơn với các điểm bất thường. **Sai số phần trăm trung bình** Hàm mất mát L2 đánh giá tổng sai số phần trăm giữa giá trị dự đoán và giá trị thật sự của điểm dữ liệu theo công thức:

$$MAPE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{\sum_{i=1}^{n} y_i} \cdot 100\%$$

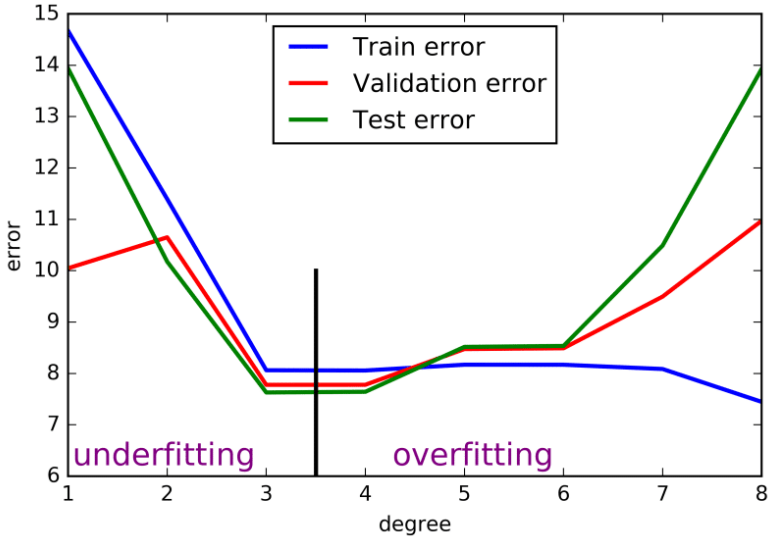
2.3.4 Vấn đề overfitting và underfitting

Statistical Fit

Statistical Fit (độ chính xác trong thống kê) là chỉ độ gần đúng của hàm xây dưng với hàm hàm mục tiêu. Các phương thức được sử dụng trong thống kê khác với phương thức thực hiện trong học máy. Ví dụ, trong thống kê thường sử dụng các phương pháp ước lượng để ước lượng hàm

mục tiêu. Tuy nhiên, trong học máy, ta lại không sử dụng phương pháp đó. Học máy dựa trên việc học từ dữ liệu, ta đưa ra mô hình xấp xỉ chính xác nhất từ bộ dữ liệu mẫu có thể có nhiễu.

Statistical Fit cũng được sử dụng trong học máy như một thước đo. Một số kỹ thuật trong thống kê cũng được áp dụng trong học máy (ví dụ: tính sai số).



Hình 2.9: Mô tả Overfitting và Underfitting

Overfitting

Overfitting là hiện tượng khi mô hình xây dựng thể hiện được chi tiết bộ dữ liệu huấn luyện. Điều này có nghĩa là cả dữ liệu nhiễu, hoặc dữ liệu bất thường trong tập huấn luyện đều được chọn và học để đưa ra quy luật mô hình. Những quy luật này sẽ không có ý nghĩa nhiều khi áp dụng với bộ dữ liệu mới có thể có dạng dữ liệu nhiễu khác. Khi đó, nó ảnh hưởng tiêu cực tới độ chính xác của mô hình nói chung.

Hiện tượng Overfitting thường xảy ra trong các mô hình phi tham số hoặc phi tuyến, những mô hình có sự linh hoạt cao trong xây dựng hàm mục tiêu.

Như vậy, rất nhiều thuật toán học máy phi tham số sẽ bao gồm những thông số và kĩ thuật để hạn chế và giới hạn múc độ học chi tiết của mô hình.

Ví dụ, bài toán cây quyết định là một thuật toán học máy phi tham số. Đây là thuật toán thường xảy ra hiện tượng Overfitting. Ta có thể tránh hiện tượng này bằng phương pháp cắt tỉa cây (pruning).

Underfitting

Underfitting (chưa khớp) là hiện tượng khi mô hình xây dựng chưa có độ chính xác cao trong tập dữ liệu huấn luyện cũng như tổng quát hóa với tổng thể dữ liệu. Khi hiện tượng Underfitting xảy ra, mô hình đó sẽ không phải là tốt với bất kì bộ dữ liệu nào trong vấn đề đang nhắc tới.
Hiện tượng Underfitting thường ít xảy ra trong bài toán hơn. Khi Underfitting xảy ra, ta có thể khắc phục bằng cách thay đổi thuật toán hoặc là bổ sung thêm dữ liệu đầu vào.

Good fitting

Good Fitting (vừa khớp) là nằm giữa Underfitting và Overfitting. Mô hình cho ra kết quả hợp lý với cả tập dữ liệu huấn luyện và các tập dữ liệu mới. Đây là mô hình lý tưởng mang được tính tổng quát và khớp được với nhiều dữ liệu mẫu và cả các dữ liệu mới.
Good Fitting là mục tiêu của mỗi bài toán. Tuy nhiên, trên thực tế, vấn đề này rất khó thực hiện. Để tìm được điểm Good Fitting, ta phải theo dõi hiệu suất của thuật toán học máy theo thời gian khi thuật toán thực hiện việc học trên bộ dữ liệu huấn luyện. Ta có thể mô tả và thể hiện các thông số mô hình, độ chính xác của mô hình trên cả hai tập dữ liệu huấn luyện và đào tạo.
Theo thời gian và theo quá trình học, sai số của mô hình trên bộ dữ liệu huấn luyện sẽ giảm xuống. Tuy nhiên, nếu quá trình training quá lâu, độ chính xác của mô hình có thể giảm do vấn đề Overfitting, và việc học sẽ thực hiện trên cả dữ liệu nhiễu và dữ liệu bất thường của bộ huấn luyện. Đồng thời, sai số với bộ dữ liệu kiểm định sẽ tăng lên do khả năng phổ quát hóa của mô hình giảm xuống.
Chúng ta kì vọng rằng tại thời điểm trước khi sai số trên bộ dữ liệu có dấu hiệu tăng lên, khi đó, mô hình là tốt nhất trên cả bộ dữ liệu huấn luyện và bộ dữ liệu kiểm định.

2.4 Các mô hình học máy được sử dụng

2.4.1 Linear Regression

Linear Regression là một mô hình kinh điển trong học máy, trong đó ta có bộ dữ liệu gồm n ví dụ $(x_1, x_2, x_3, ..., x_n)$ và giá trị cần dự đoán $(y_1, y_2, y_3, ..., y_n)$. Mục tiêu của mô hình là đưa ra một hàm dự đoán tuyến tính $\hat{y} = f(x)$ mà giá trị dự đoán $\hat{y}$ gần với y. Hàm đó có dạng:

$$f(x) = \theta' x + \theta_0$$

Ta muốn hàm f càng khớp với bộ dữ liệu càng tốt. Nghĩa là sai số dự đoán $\hat{y_i}$ và y_i trên điểm dữ liệu x_i càng nhỏ càng tốt. Chúng ta xây dựng hàm mất mát $J(\theta)$ là trung bình của tổng bình phương giá trị sai lệch trên từng điểm dữ liệu trên tập huấn luyện.

$$J(\theta) = \frac{1}{2}\sum(y_i - \hat{y_i})^2$$

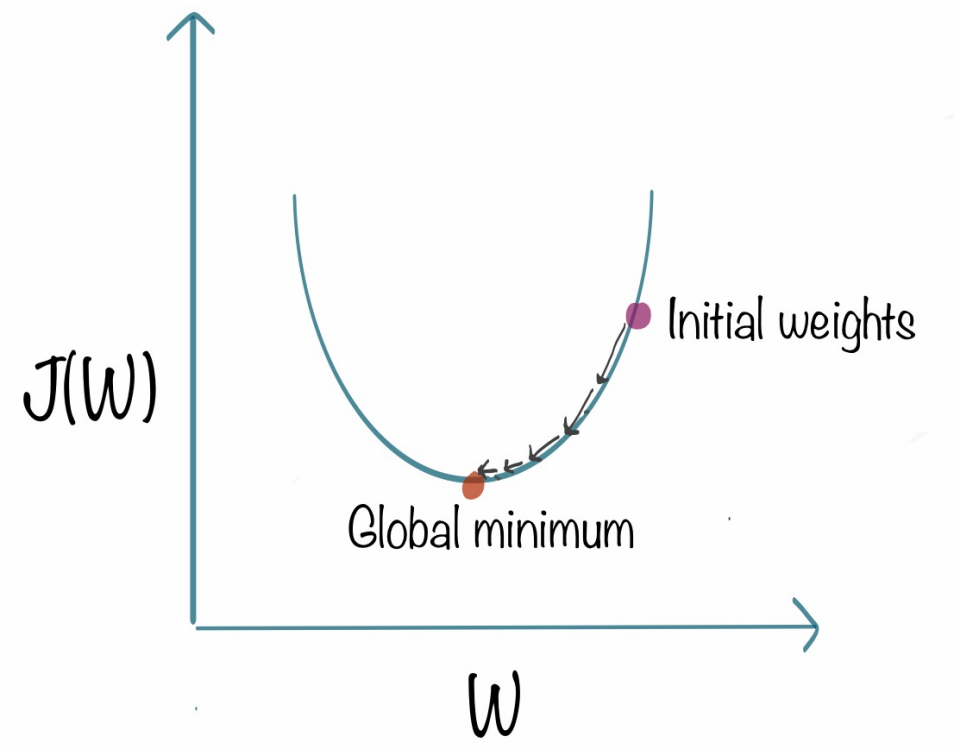
Để mô hình dự đoán càng gần với các điểm dữ liệu trên tập huấn luyện, ta cần tối thiểu hóa hàm lỗi trên, dẫn đến bài toán tối ưu sau:

$$\min J(\theta) = \frac{1}{2}\sum(y_i - \hat{y_i})^2$$

Khi đó nghiệm của bài toán là:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} J(\theta)$$

Để giải bài toán tối ưu này, ta sử dụng phương pháp Gradient Descent. Phương pháp Gradient Descent là một phương pháp tối ưu hóa, sử dụng phương pháp lặp để tiến dần về điểm tối ưu.



Hình 2.10: Giải thuật Gradient Descent

Ta cần tìm tham số $\theta \in R^n$ để tối thiểu hóa hàm lỗi $J(\theta)$. Đầu tiên, ta đặt θ tại một điểm bất kì nào đó. Sau đó, giải thuật Gradient Descent được thực hiện bằng cách cập nhật dần các tham số θ ngược với hướng của gradient $\nabla_\theta J(\theta)$ tại điểm hiện tại cho tới khi nó hội tụ về điểm nhỏ nhất. Tại mỗi bước cập nhật, ta sẽ dịch tham số đi một lượng $\alpha\nabla_\theta J(\theta)$ với tốc độ học $\alpha > 0$ thể hiện mức độ di chuyển của tham số. Như vậy sau bước thứ k, giá trị θ được xác định bởi:

$$\theta^{(k+1)} = \theta^{(k)} - \alpha\nabla_\theta J(\theta)$$

Quá trình cập nhật giá trị của *theta* dừng lại nếu gặp một trong các điều kiện sau:

- Khi số lần cập nhật đã vượt quá giới hạn số lần cập nhật đã được quy ước bởi người cài đặt giải thuật.
- Kiểm tra giá trị của hàm lỗi, nếu sau 2 lần cập nhật liên tiếp, giá trị hàm lỗi không giảm nhiều thì ta có thể coi như đã hội tụ về điểm cực trị và dừng giải thuật.

2.4.2 Support Vector Regression

Giới thiệu

Support Vector Machine là một công cụ học máy phổ biến cho các bài toàn phân loại và hồi quy, được phát minh bởi Vladimir Vapnik và cộng sự vào năm 1992. SVM regression được coi là một kỹ thuật nonparametric vì nó dựa vào hàm kernel. Giải thuật SVR còn được gọi là episilon-insensitive SVM(ε-SVM), hay L1-loss. Trong giải thuật ε-SVM, tập các điểm dữ liệu huấn luyện bao gồm các ví dụ x_n và giá trị đích y_n. Mục tiêu là tìm một hàm f sao cho $f(x_n)$ sai khác với y_n một khoảng nhỏ hơn ε.

Linear Support Vector Regression

Dạng chính tắc

Giả sử ta có một tập *n* điểm dữ liệu huấn luyện $(x_1^m, x_2^m, x_3^m, ..., x_n^m)$ trong đó mỗi điểm dữ liệu là một vector *m* chiều và tập các giá trị $(y_1, y_2, y_3, ..., y_n)$ tương ứng. Mục tiêu của thuật toán là tìm một hàm tuyến tính f sao cho giá trị $f(x_i)$ sai khác không quá ε so với giá trị y_i:

$$f(x) = x'\beta + b$$

Nó được phát biểu dưới dạng một bài toán tối ưu lồi như sau:

$$\min J(\beta) = \frac{1}{2}\beta'\beta$$

Thỏa mãn:

$$\forall i : |y_i - (x_i'\beta + b)| \leq \varepsilon$$

Để đảm bảo luôn tìm được một hàm thỏa mãn điều kiện trên với mọi điểm dữ liệu, ta thêm biến bù ξ_i và ξ_i^*. Phương pháp này gọi là Soft Margin Support Vector Regression, do biến bù cho phép sai số dự đoán tồn tại, có giá trị trong khoảng ξ_i và ξ_i^* vẫn thỏa mãn điều kiện được yêu cầu.

Thêm các biến bù dẫn tới hàm mục tiêu ở dạng chính tắc như sau:

$$\min J(\beta) = \frac{1}{2}\beta'\beta + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

Thỏa mãn:

$$\begin{gathered}\forall i : |y_i - (x_i'\beta + b)| \leq \varepsilon + \xi_i \\ \forall i : |(x_i'\beta + b) - y_i| \leq \varepsilon + \xi_i^* \\ \forall i : \xi_i \geq 0 \\ \forall i : \xi_i^* \geq 0\end{gathered}$$

C là một hằng số dương kiểm soát giá trị của hàm mục tiêu, C càng lớn thì các ví dụ trong tập training nằm ngoài epsilon margin(ε) gây tăng giá trị cho hàm mục tiêu. Từ đó, C ngăn chặn overfitting.

Hàm mất mát trong trường hợp này bỏ qua các lỗi nằm trong khoảng ε và coi như chúng bằng 0 và được biểu diễn bởi:

$$L_\varepsilon = \begin{cases} 0 & \text{với } | y - f(x) | \leq \varepsilon \\ | y - f(x) | - \varepsilon & \text{với các trường hợp còn lại} \end{cases}$$

Dạng đối ngẫu

Bài toán tối ưu nêu ở phần trên có thể giải quyết với chi phí tính toán thấp hơn ở dạng đối ngẫu Lagrange. Nghiệm của dạng đối ngẫu và dạng chính tắc không nhất thiết phải giống nhau, và sự sai khác giữa chúng gọi là duality gap. Khi bài toán tối ưu lồi và thỏa mãn điều kiện ràng buộc hợp cách (constrain qualification condition), nghiệm của dạng chính tắc chính là nghiệm của dạng đối ngẫu.

Ta thêm vào nhân tử Lagrange không âm α_n và α_n^* với mỗi điểm dữ liệu x_n để đạt được dạng đối ngẫu, khi đó ta tối thiểu hóa hàm

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i'x_j + \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i)$$

Thỏa mãn:

$$\sum_{n=1}^{N}(\alpha_n - \alpha_n^*) = 0$$
$$\forall n : 0 \leq \alpha_n \leq C$$
$$\forall n : 0 \leq \alpha_n^* \leq C$$

Khi đó, β có thể được xác định bằng tổ hợp tuyến tính của các điểm dữ liệu huấn luyện bằng công thức

$$\beta = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)x_n$$

Hàm xác định giá trị đích phụ thuộc vào các support vectors:

$$f(x) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)(x_n'x) + b$$

Điều kiện Karush-Kuhn-Tucker(KKT) phải được thỏa mãn để đảm bảo nghiệm tối ưu. Với bài toán trên, các ràng buộc này là:

$$\forall n : \alpha_n(\varepsilon + \xi_n - y_n + x_n'\beta + b) = 0$$
$$\forall n : \alpha_n^*(\varepsilon + \xi_n^* - y_n + x_n'\beta - b) = 0$$
$$\forall n : \xi_n(C - \alpha_n) = 0$$
$$\forall n : \xi_n^*(C - \alpha_n^*) = 0$$

Điều kiện KKT chỉ ra rằng các điểm dữ liệu nằm hoàn toàn trong khoảng ε-margin có $\alpha_n = 0$ và $\alpha_n^* = 0$. Nếu hoặc α_n, α_n^* khác 0, những điểm dữ liệu đó gọi là các support vector.

Non-linear Support Vector Regression

Trong thực tế, các bài toán hồi quy không phải lúc nào cũng có một lời giải với các mô hình tuyến tính. Trong trường hợp đó, dạng đối ngẫu Lagrange cho phép đưa ra mô hình phi tuyến, bằng cách thay thế tích $x_1'x_2$ bằng một hàm kernel phi tuyến $G(x_1, x_2) = \langle\varphi(x_1), \varphi(x_2)\rangle$, với $\varphi(x)$ là một ánh xạ biến đổi x lên một không gian nhiều chiều hơn. Các hàm kernel thông dụng là:

- Hàm tuyến tính:

$$G(x_j, x_k) = x_j'x_k$$

- Gaussian kernel(Radius Basis Function):

$$G(x_j, x_k) = \exp(\frac{||x_j - x_k||^2}{2\delta^2})$$

- Hàm đa thức(Polynomial Kernel):

$$G(x_j, x_k) = (1 + x_j'x_k)^q$$

Ma trận Gram là ma trận cỡ $n \times n$ chứa các phần tử $g_{\mathrm{i,j}} = G(x_i, x_j)$. Mỗi phần tử $g_{\mathrm{i,j}}$ bằng với tích vô hướng trên không gian mới. Tuy vậy, chúng ta không cần biết chính xác hàm φ, vì ta có thể dùng hàm kernel để tạo ra ma trận Gram trực tiếp.

Dạng đối ngẫu cho Support Vector Regression phi tuyến thay thế tích vô hướng $(xi'xj)$ bằng phần tử tương ứng $(g_{\mathrm{i,j}})$ trong ma trận Gram. Khi đó, hàm mục tiêu cần tối ưu ở dạng đối ngẫu là:

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)G(x_i, x_j) + \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i)$$

Thỏa mãn:

$$\sum_{n=1}^{N}(\alpha_n - \alpha_n^*) = 0$$
$$\forall n : 0 \le \alpha_n \le C$$
$$\forall n : 0 \le \alpha_n^* \le C$$

Khi đó:

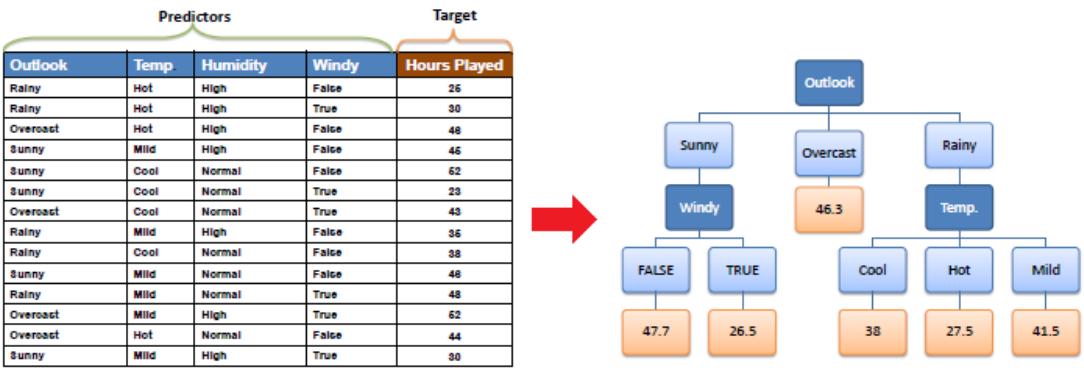
$$f(x) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)G(x_n, x) + b$$

Điều kiện KKT cần thỏa mãn:

$$\forall n : \alpha_n(\varepsilon + \xi_n - y_n + f(x_n)) = 0$$
$$\forall n : \alpha_n^*(\varepsilon + \xi_n^* + y_n - f(x_n)) = 0$$
$$\forall n : \xi_n(C - \alpha_n) = 0$$
$$\forall n : \xi_n^*(C - \alpha_n^*) = 0$$

2.4.3 Decision Tree Regression

Cây quyết định được sử dụng để xây dựng cả mô hình hồi quy hoặc phân loại với mô hình dạng cây. Nó chia nhỏ một tập dữ liệu thành các tập nhỏ hơn, trong khi đồng thời từng bước xây dựng cây quyết định tương ứng. Kết quả là là một mô hình cây với các nút quyết định và nút lá. Một nút quyết định có tối thiểu 2 nhánh, mỗi nhánh đại diện cho giá trị của thuộc tính được kiểm tra tại nút đó. Nút lá đại diện cho giá trị mục tiêu, có thể là cả dạng nhãn(bài toán phân loại), hoặc giá trị số(bài toán hồi quy).



Hình 2.11: Giải thuật Decision Tree

Giải thuật chính để xây dựng cây quyết định là ID3, C4.5, và CART. Ở đây, ta khảo sát giải thuật ID3, là một giải thuật được phát triể bởi J.R.Quinlan, xây dựng cây theo mô hình top-down, tìm kiếm tham lam trên không gian các nhánh có thể phân chia và không quay lui. Trong khi giải thuật ID3 cho bài toán phân loại sử dụng độ đo Information Gain, thì với bài toán hồi quy ID3 sử dụng độ đo sự suy giảm độ lệch chuẩn(standard deviation reduction) để xây dựng cây.

Độ lệch chuẩn

Ta sử dụng độ lệch chuẩn để tính độ nhất quán trong một tập các điểm dữ liệu. Nếu các điểm này hoàn toàn đồng nhất, giống hệt nhau, thì độ lệch chuẩn trên tập này bằng 0.
Độ lệch chuẩn trên một thuộc tính
Giá trị trung bình:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{N} x_i$$

Độ lệch chuẩn:

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Hệ số biến thiên:

$$CV = \frac{s}{\bar{x}} \cdot 100\%$$

Ví dụ:

Hours Played
25
30
46
45
52
23
43
35
38
46
48
52
44
30

$\Rightarrow$

$$\textbf{Count} = n = 14$$

$$\textbf{Average} = \bar{x} = \frac{\sum x}{n} = 39.8$$

$$\textbf{Standard Deviation} = S = \sqrt{\frac{\sum(x-\bar{x})^2}{n}} = 9.32$$

$$\textbf{Coeffeicient of Variation} = CV = \frac{S}{\bar{x}} * 100\% = 23\%$$

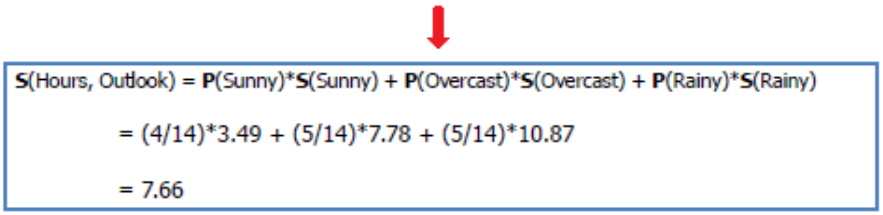
Độ lệch chuẩn trên giá trị dự đoán và một thuộc tính

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

Trong đó $P(c)$ là giá trị trung bình của giá trị đích khi thuộc tính đó bằng c.
Ví dụ:

$$S(T,X) = \sum_{c \in X} P(c)S(c)$$

		Hours Played (StDev)	Count
Outlook	Overcast	3.49	4
	Rainy	7.78	5
	Sunny	10.87	5
			14



Huấn luyện cây quyết định: Standard Deviation Reduction

Xây dựng cây quyết định là việc tìm các thuộc tính mà ta có thể tìm được luật phân chia trên thuộc tính đó, sao cho độ lệch chuẩn trên tập dữ liệu giảm nhiều nhất.
Mã giả thực hiện giải thuật cây quyết định:

1. Khởi tạo 1 nút, gán toàn bộ tập dữ liệu huấn luyện vào nút hiện tại, đặt nút đó làm nút gốc, tính độ lệch chuẩn của toàn bộ dữ liệu trên giá trị đích.
2. Với mỗi thuộc tính, phân tách dữ liệu ở nút đó bằng giá trị của thuộc tính đó. Sau đó tính giá trị giảm độ lệch chuẩn Standard Deviation Reduction.
3. Xác định thuộc tính cho độ giảm độ lệch chuẩn nhiều nhất trên nút hiện tại, đặt thuộc tính đó là thuộc tính phân chia cho nút hiện tại, và phân chia bộ dữ liệu trong nút hiện tại theo lát cắt đã xác định.
4. Đặt các bộ dữ liệu đã phân chia tại nút hiện tại là một nút cần phân chia tiếp.
5. Lặp lại bước 2 cho đến khi độ sâu cây vượt quá tham số điều khiển độ sâu, hoặc số điểm dữ liệu trong một nút nằm trong khoảng đã định trước, hoặc giá trị CV nhỏ hơn điều kiện dừng quy ước.

Trong ví dụ trên, các bước được thực hiện như sau:
Bước 1: Khởi tạo nút gốc, gán toàn bộ tập dữ liệu vào nút đó, tính độ lệch chuẩn $s = 9.32$
Bước 2: Bộ dữ liệu được chia bằng các thuộc tính khác nhau, và tính độ lệch chuẩn trên từng nhánh. Giá trị giảm độ lệch chuẩn được tính bằng hiệu của độ lệch chuẩn trước khi chia và độ lệch chuẩn sau khi phân chia.

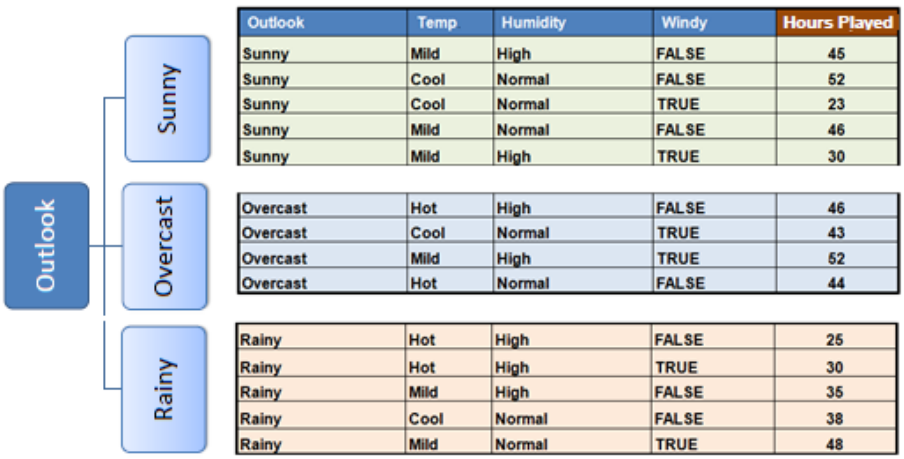
		Hours Played (StDev)
Outlook	Overcast	3.49
	Rainy	7.78
	Sunny	10.87
SDR=1.66		

		Hours Played (StDev)
Temp.	Cool	10.51
	Hot	8.95
	Mild	7.65
SDR=0.17		

		Hours Played (StDev)
Humidity	High	9.36
	Normal	8.37
SDR=0.28		

		Hours Played (StDev)
Windy	False	7.87
	True	10.59
SDR=0.29		

Bước 3: Chọn thuộc tính Outlook có giá trị giảm độ lệch chuẩn cao nhất.
Bước 4: Bộ dữ liệu được chia dựa trên thuộc tính đã chọn ở bước 3. Quá trình này được chạy đệ quy với các nút không phải nút lá, cho đến khi thỏa mãn điều kiện dừng.



Outlook	Temp	Humidity	Windy	Hours Played
Sunny	Mild	High	FALSE	45
Sunny	Cool	Normal	FALSE	52
Sunny	Cool	Normal	TRUE	23
Sunny	Mild	Normal	FALSE	46
Sunny	Mild	High	TRUE	30

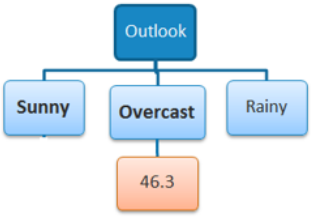
Overcast	Hot	High	FALSE	46
Overcast	Cool	Normal	TRUE	43
Overcast	Mild	High	TRUE	52
Overcast	Hot	Normal	FALSE	44

Rainy	Hot	High	FALSE	25
Rainy	Hot	High	TRUE	30
Rainy	Mild	High	FALSE	35
Rainy	Cool	Normal	FALSE	38
Rainy	Mild	Normal	TRUE	48

Bước 5a: Với giá trị của Outlook="Overcast", bộ dữ liệu đã được phân chia vào phần này không cần chia thêm vì $CV(8\%)$ nhỏ hơn ngưỡng 10%. Nút lá này nhận giá trị trung bình của các mẫu nằm trong nút đó.

Outlook - Overcast

		Hours Played (StDev)	Hours Played (AVG)	Hours Played (CV)	Count
Outlook	Overcast	3.49	46.3	8%	4
	Rainy	7.78	35.2	22%	5
	Sunny	10.87	39.2	28%	5



Bước 5b: Với nhánh Outlook="Sunny", CV(28%) lớn hơn ngưỡng nên cần phải chia ra thành các nhánh con. Ta chọn thuộc tính Windy là tiêu chuẩn phân chia tại nút này vì nó cho suy giảm độ lệch chuẩn lớn nhất.

Outlook - Sunny

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Cool	Normal	TRUE	23
Mild	Normal	FALSE	46
Mild	High	TRUE	30
			S = 10.87
			AVG = 39.2
			CV = 28%

		Hours Played (StDev)	Count
Temp	Cool	14.50	2
	Mild	7.32	3

SDR = 10.87-((2/5)*14.5 + (3/5)*7.32) = **0.678**

		Hours Played (StDev)	Count
Humidity	High	7.50	2
	Normal	12.50	3

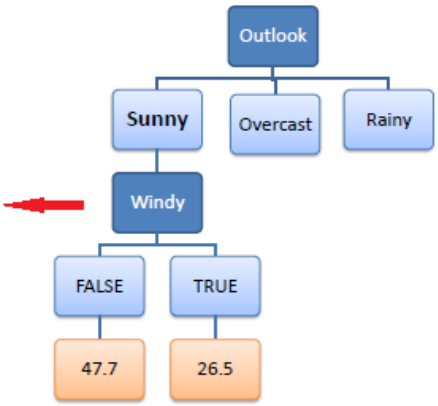
SDR = 10.87-((2/5)*7.5 + (3/5)*12.5) = **0.370**

		Hours Played (StDev)	Count
Windy	False	3.09	3
	True	3.50	2

SDR = 10.87-((3/5)*3.09 + (2/5)*3.5) = **7.62**

Với cả nhánh Windy=True hoặc nhánh Windy=False, số điểm dữ liệu đều nhỏ hơn 3, nên ta dừng việc chia nhánh và gán giá trị trung bình của các điểm dữ liệu vào các nút lá tương ứng.

Temp	Humidity	Windy	Hours Played
Mild	High	FALSE	45
Cool	Normal	FALSE	52
Mild	Normal	FALSE	46
Cool	Normal	TRUE	23
Mild	High	TRUE	30



Bước 5c: Nhánh Outlook="Rainy" có CV(22%) lớn hơn ngưỡng, cần phải phân chia. Ta chọn Temp là thuộc tính phân chia tại nút này, do có suy giảm độ lệch chuẩn lớn nhất

Outlook - Rainy

Temp	Humidity	Windy	Hours Played
Hot	High	FALSE	25
Hot	High	TRUE	30
Mild	High	FALSE	35
Cool	Normal	FALSE	38
Mild	Normal	TRUE	48
			S = 7.78
			AVG = 35.2
			CV = 22%

		Hours Played (StDev)	Count
Temp	Cool	0	1
	Hot	2.5	2
	Mild	6.5	2

SDR = 7.78 - ((1/5)*0+(2/5)*2.5 + (2/5)*6.5) = **4.18**

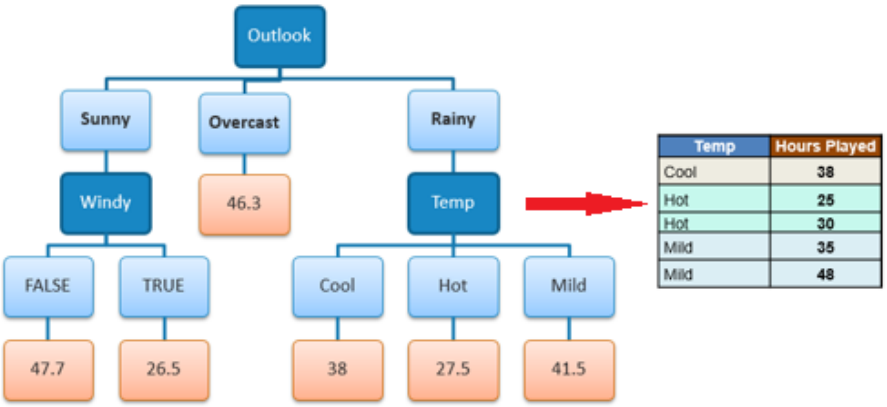
		Hours Played (StDev)	Count
Humidity	High	4.1	3
	Normal	5.0	2

SDR = 7.78 - ((3/5)*4.1 + (2/5)*5.0) = **3.32**

		Hours Played (StDev)	Count
Windy	False	5.6	3
	True	9.0	2

SDR = 7.78 - ((3/5)*5.6 + (2/5)*9.0) = **0.82**

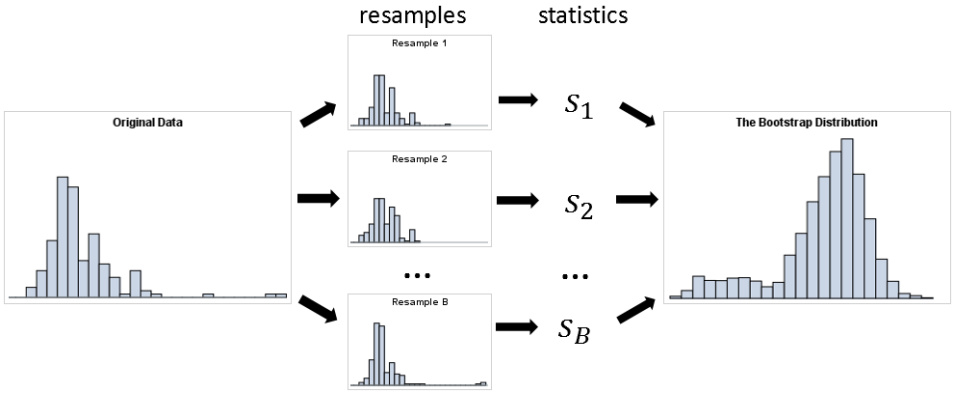
Các nút được phân chia tại nút này là nút lá, vì chúng chỉ chứa số điểm dữ liệu nhỏ hơn 3. Đến đây ta dừng thuật toán và đưa ra mô hình cây quyết định như sau:



2.4.4 Random Forest Regression

Bootstrap

Bootstrap là một phương pháp nổi tiếng trong thống kê được giới thiệu bởi Bradley Efron vào năm 1979. Phương pháp này chủ yếu dùng để ước lượng lỗi chuẩn(standard errors), độ lệch(bias) và tính toán khoảng tin cậy cho các tham số. Phương pháp này được thực hiện như sau: từ một quần thể ban đầu lấy ra một mẫu $L = (x_1, x_2, ..., x_n)$ gồm n thành phần, tính toán các tham số mong muốn. Trong các bước tiếp theo lặp lại b lần việc tạo ra mẫu L_b cũng gồm n phần tử từ L bằng cách lấy lại mẫu với sự thay thế các thành phần trong mẫu ban đầu sau đó tính toán lại các tham số mong muốn.



Hình 2.12: Phương pháp Bootstrap

Bagging

Bagging là từ viết tắt của Bootstrap aggregating, là phương pháp tổng hợp kết quả từ các bootstrap. Tư tưởng chính của phương pháp này như sau: Cho một tập huấn luyện $D = |(x_i, y_i) : i = 1, 2, , ..., n|$ và giả sử chúng ta muốn có một dự đoán nào đó với biến x.

- Một mẫu gồm B tập dữ liệu, mỗi tập dữ liệu gồm n phần tử được lựa chọn ngẫu nhiên từ D với sự thay thế. Do đó $B = (D_1, D_2, ..., D_n)$ trông giống như là một tập các tập huấn luyện được nhân bản
- Huấn luyện một mô hình đối với mỗi tập D_b và lần lượt thu được các dự đoán trên tập D_b
- Kết quả cuối cùng được tính trung bình dự đoán của các mô hình với bài toán hồi quy và bỏ phiếu, chọn nhãn được bỏ phiếu nhiều nhất với bài toán phân loại

Phương pháp bagging có thể làm giảm phương sai:

- Nếu mỗi bộ mô hình đơn lẻ không ổn định có nghĩa là mô hình đó có phương sai lớn, mô hình tổng hợp sẽ có phương sai nhỏ hơn so với một mô hình đơn lẻ.
- Một mô hình tổng hợp có thể xem như một xấp xỉ tới giá trị trung bình tật sự có được bằng cách thay đổi các phân bố xác suất và các bootstrap.

Random Forest

Giải thuật Random Forest tăng độ chính xác của mô hình bằng cách sử dụng nhiều cây quyết định, bằng kĩ thuật bootstraping và bagging. Khi ra quyết định, mô hình Random Forest sẽ lấy trung bình giá trị dự đoán trên các cây trong rừng. Ý tưởng chính của giải thuật như sau:

- Ở mỗi lần phân chia cây, một tập ngẫu nhiên m thuộc tính được lấy ra và chỉ m thuộc tính này tham gia vào việc phân chia cây. Thông thường, $m = \sqrt{p}$ hoặc $m = \log_2 p$, trong đó p là tổng số các thuộc tính.
- Đối với mỗi cây phát triển dựa vào một mẫu bootstrap, tỷ lệ lỗi của các phần tử không thuộc vào bootstrap được kiểm soát, gọi là tỷ lệ lỗi out-of-bag(OOB).

Dữ liệu out-of-bag được sử dụng để ước lượng lỗi tạo ra từ việc kết hợp các kết quả từ các cây, tổng hợp trong giải thuật Random Forest, cũng như để ước lượng độ quan trọng của thuộc tính.

2.5 Các nền tảng công nghệ được sử dụng

2.5.1 Ngôn ngữ lập trình Python

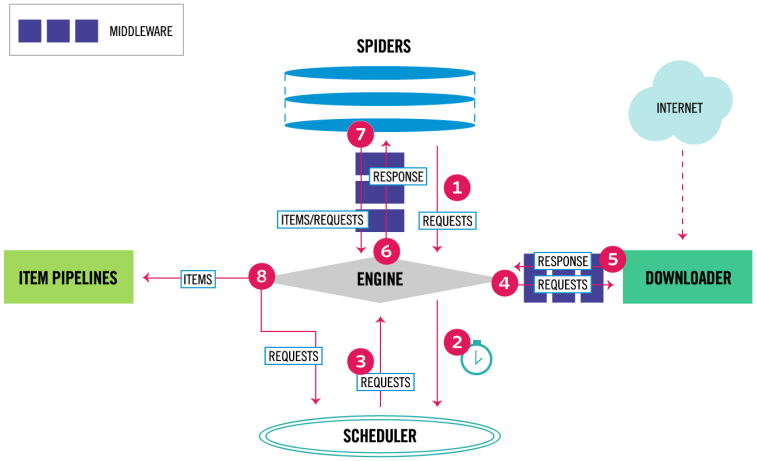
Python là một ngôn ngữ lập trình thông dịch, kiểu động, mã nguồn mở. Là một ngôn ngữ lập trinh tinh gọn, dễ học, nhưng đầy sức mạnh, hỗ trợ tốt nhiều tác vụ với hệ thống thư viện của bên thứ ba.
Các ứng dụng của ngôn ngữ lập trình Python:

- Phát triển ứng dụng web
- Tính toán khoa học
- Lập trình kịch bản hệ thống, tự động tác vụ
- Phát triển phần mềm giao diện

2.5.2 Framework thu thập dữ liệu Scrapy

Scrapy là một nền tảng thu thập dữ liệu hiệu năng cao, dễ sử dụng. Nó được dùng để lấy dữ liệu có cấu trúc từ các website, khai phá dữ liệu, monitor hệ thống và kiểm thử tự động. Được xây dựng dựa trên thư viện lập trình bất đồng bộ Twisted, Scrapy làm việc với hiệu năng cao với các tác vụ cần network I/O.

Kiến trúc tổng quan



Hình 2.13: Kiến trúc framework thu thập dữ liệu Scrapy

Các thành phần chính của Scrapy

- Spider: định nghĩa tập luật khai phá và trích rút dữ liệu.
- Engine: điều phối và trung chuyển thông điệp giữa các thành phần.
- Scheduler: lập lịch cho các yêu cầu để đưa ra Downloader.
- Downloader: thực hiện các HTTP Request, chuyển cho engine để đưa đến spider để trích rút dữ liệu.
- Item pipelines: xử lý các dữ liệu được trích rút từ spider, các task cụ thể như là làm sạch, kiểm tra, lưu trữ vào cơ sở dữ liệu hoặc ghi ra file.D

Luồng xử lý của Framework Scrapy

1. Spider gửi request đến cho engine, yêu cầu engine lập lịch cho request
2. Engine lập lịch cho request ở trong bộ scheduler
3. Scheduler trả về Request đã lập lịch về cho Engine
4. Engine gửi Request cho Downloader, yêu cầu download.
5. Downloader download webpage đó, tạo ra Response object, gửi về Engine
6. Engine nhận được Response, sau đó gửi về cho Spider để trích rút dữ liệu
7. Spider trích rút dữ liệu xong, tạo ra 1 Item, gửi lại cho Engine
8. Engine gửi Item đó cho Item Pipelines để xử lý dữ liệu
9. Quá trình trên lặp lại từ bước 1 cho đến khi không còn Request đến Engine.

2.5.3 Các thư viện hỗ trợ xử lý, trực quan hóa dữ liệu

Numpy

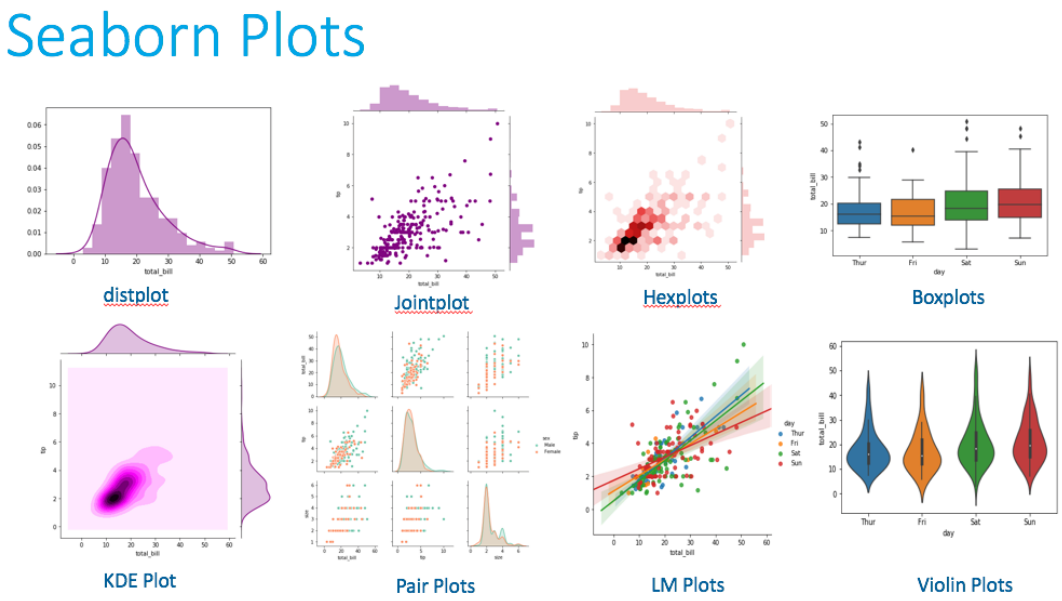
Numpy là thư viện tính toán số học, được viết bằng C++ để đạt được hiệu năng cao. Numpy cung cấp binding sang ngôn ngữ Python để tiện cho người lập trình sử dụng. Hỗ trợ mạnh mẽ các phép toán Đại số tuyến tính, Numpy hỗ trợ đắc lực cho việc triển khai các thuật toán song song, hiệu năng cao với Python.

Pandas

Pandas là một thư viện Python cho phân tích dữ liệu. Nó được dựng xung quanh một dữ cấu trúc dữ liệu được gọi là DataFrame, nó được mô hình hóa sau R DataFrame. Một Pandas DataFrame là một bảng, giống như một bảng tính Excel, đặc biệt nó cho phép tuy vấn tương tự SQL và 'join' các bảng. Một công cụ giá trị khác cung cấp bởi Pandas là khả năng nhập từ rất nhiều định dạng file và cơ sở dữ liệu, như SQL, Excel files, file mà các giá trị được phân tách bởi dấu phảy (CSV).

Seaborn

Seaborn là một thư viện mã nguồn mở, được xây dựng để đơn giản hóa trực quan hóa dữ liệu. Seaborn chủ yếu cung cấp các hàm vẽ đồ thị mô tả, thống kê dữ liệu.



Hình 2.14: Các biểu đồ thống kê được hỗ trợ bởi Seaborn

2.5.4 Thư viện Học máy Scikit-learn

Scikit-learn là một dự án open-source, có nghĩa là scikit-learn miễn phí để sử dụng và phân tán, bất kỳ ai cũng dễ dàng nhận được source code để xem bên trong nó hoạt động như thế nào. Scikit-learn liên tục được phát triển và cải thiện, và có một cộng đồng người sử dụng rất lớn. Nó bao gồm nhiều thuật toán học máy hiện đại nhất, cũng như tài liệu bao quát cho mỗi thuật toán trên website [http://scikit-learn.org/stable/documentation].
Scikit-learn là một công cụ rất phổ biến, và là thư viện Python nổi bật cho học máy. Nó được

sử dụng rộng rãi trong công nghiệp và học thuật, và có rất nhiều bài hướng dẫn và code và scikit-learn sẵn sàng trên Internet. Scikit-learn hoạt động hiệu quả với số lượng lớn các công cụ Python về khoa học khác.

2.5.5 Thư viện phát triển Web API Flask

Flask là một thư viện phát triển ứng dụng Web, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các API nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp các công cụ, các thư viện và các công nghệ hỗ trợ làm những công việc trên.
Flask là một micro-framework, nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

2.5.6 Tổng hợp các thư viện và nền tảng công nghệ được sử dụng trong đồ án

Thư viên, Framework	Ứng dụng
Scrapy	Thu thập dữ liệu
Numpy	Xử lý dữ liệu
Pandas	Xử lý dữ liệu
Seaborn	Trực quan hóa dữ liệu
Scikit-learn	Xây dựng, đánh giá mô hình học máy
Flask	Triển khai ứng dụng dưới dạng JSON Web API

Bảng 2.4: Tổng hợp các thư viện nguồn mở sử dụng trong đồ án

Chương 3

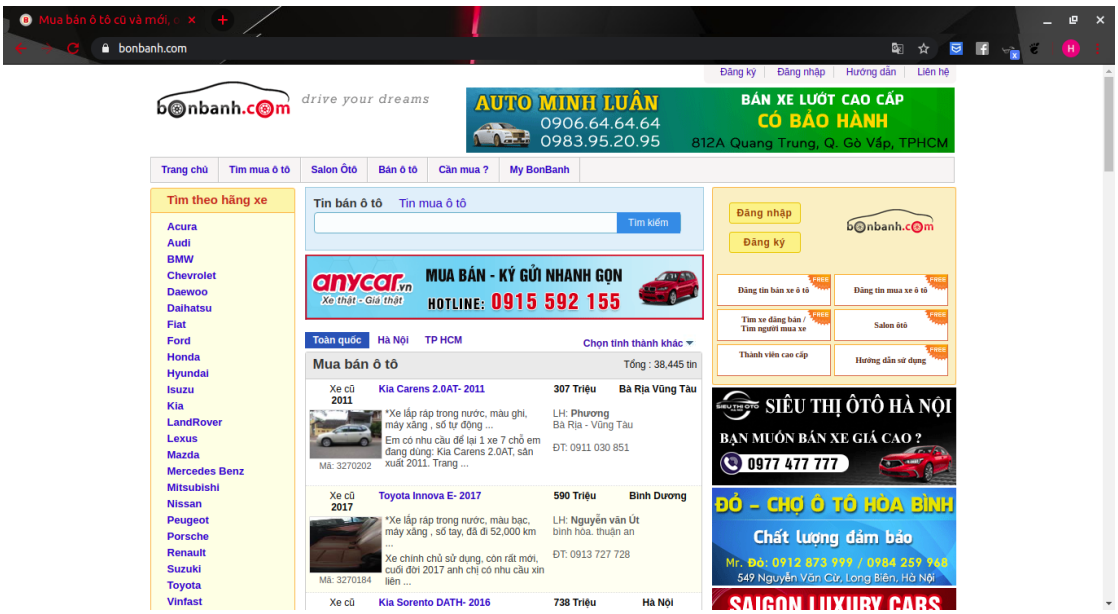
XÂY DỰNG, TRIỂN KHAI HỆ THỐNG ĐỊNH GIÁ TỰ ĐỘNG

Trong chương này, em sẽ mô tả chi tiết cách thức thực hiện các gói công việc đã đề ra. Cụ thể phần 3.1 trình bày về xây dựng trình thu thập dữ liệu, phần 3.2 trình bày về xử lý, trực quan hóa dữ liệu. Còn hai phần sau của chương lần lượt trình bày về xây dựng, thử nghiệm và triển khai mô hình Học Máy dưới dạng JSON Web API.

3.1 Xây dựng trình thu thập dữ liệu

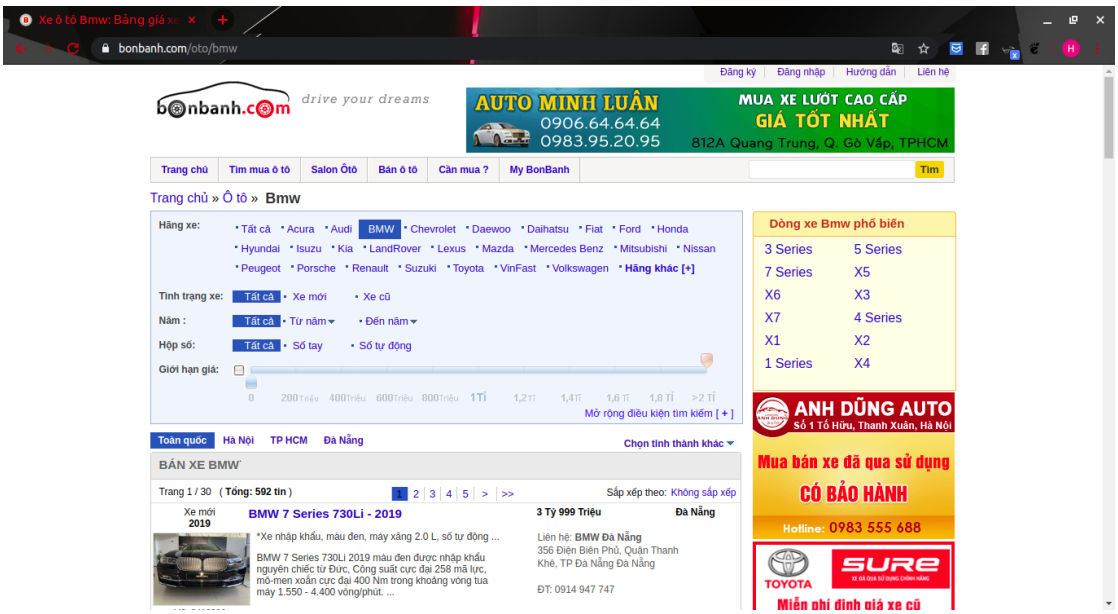
3.1.1 Đánh giá trang web cần thu thập dữ liệu

Em chọn trang web giao dịch ô tô cũ bonbanh.com để thu thập dữ liệu, đây là một sàn giao dịch lâu đời, uy tín, có lượng người dùng và lượng tương tác cao. Các xe được rao bán đều được người bán tự định giá, phong phú, đa dạng về chủng loại, mẫu mã, đời xe. Đây là một nguồn dữ liệu tốt để so sánh, tham khảo khi định giá sản phẩm xe cũ.



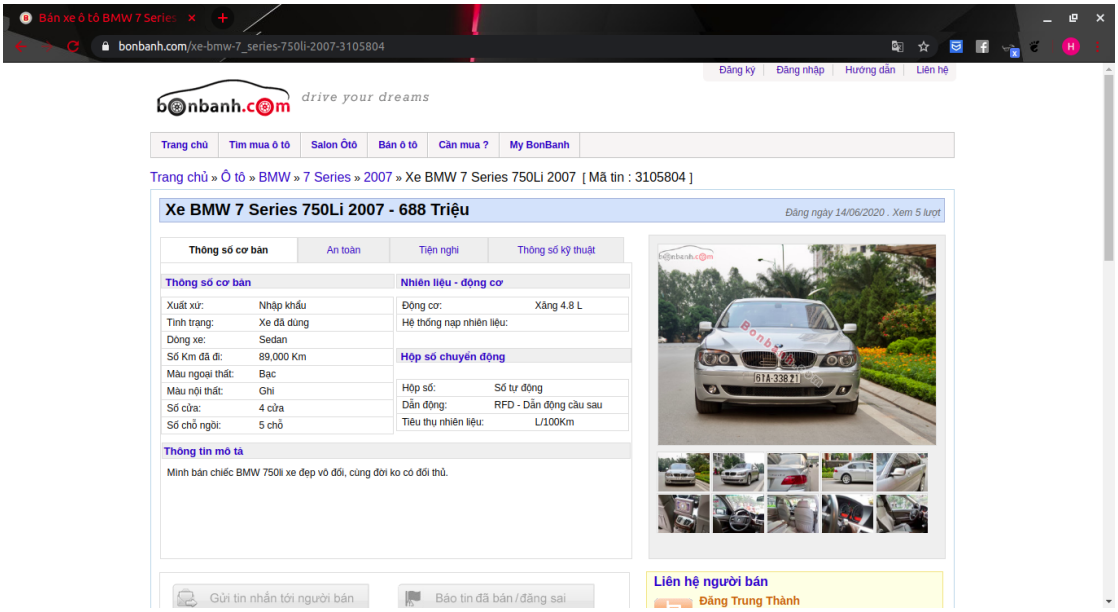
Hình 3.1: Trang chủ trang web cần thu thập dữ liệu

Trang web cung cấp một số tính năng như lọc sản phẩm theo một hoặc một vài tiêu chí nào đó.



Hình 3.2: Trang tìm kiếm, lọc thông tin

Trang thông tin rao bán xe gồm nhiều thông tin chi tiết về xe, giá bán, thông tin chủ sở hữu và số điện thoại.

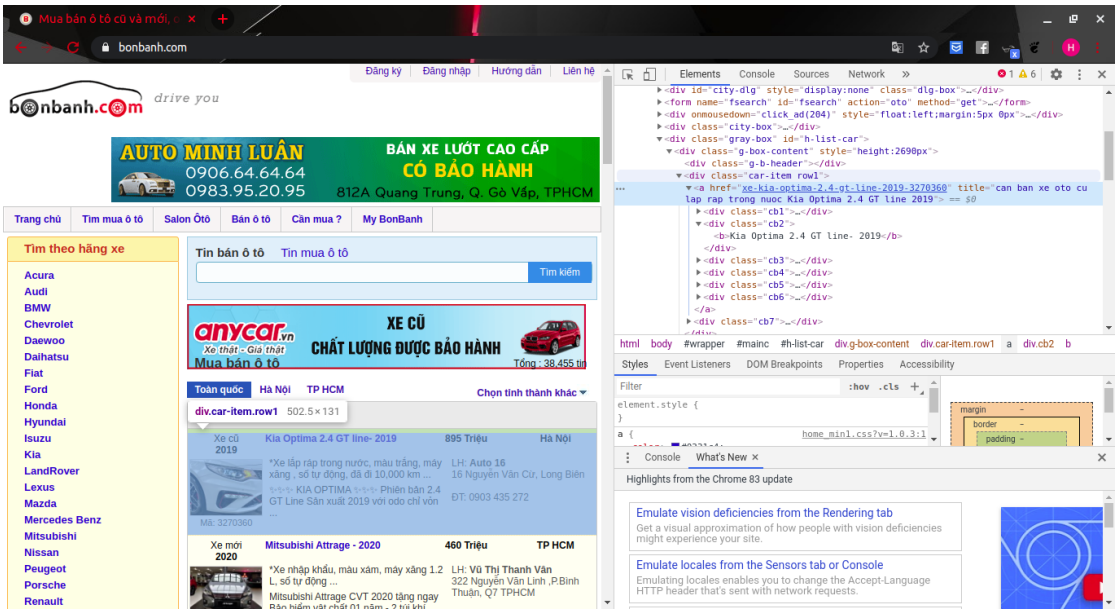


Hình 3.3: Trang thông tin chi tiết sản phẩm

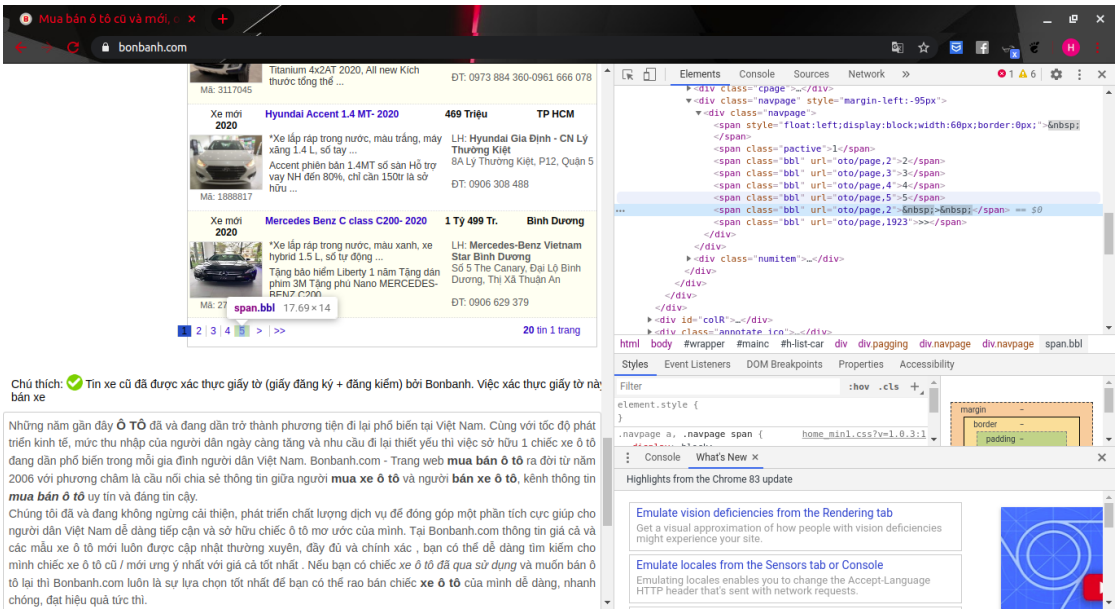
3.1.2 Xây dựng luật khai phá cho trình thu thập dữ liệu

Trang web bonbanh.com có sitemap như đã trình bày ở phần trên, người dùng có thể vào trang chủ và tùy chọn sản phẩm, xem chi tiết sản phẩm. Trình thu thập dữ liệu giả lập lại hành động trên, với tập luật khai phá sitemap giống như người dùng. Đầu tiên, trình thu thập dữ liệu vào trang chủ, lựa chọn toàn bộ sản phẩm. Sau đó, nó vào trang tin từng sản phẩm để thu thập thông tin của các sản phẩm.

Các web page được liên kết với nhau bằng thẻ <a>, trong đó có thuộc tính href dùng để trỏ vào trang web đích. Khi người dùng click vào thẻ <a> sẽ được điều hướng đến trang web đích mà URL nằm trong thuộc tính href.



Hình 3.4: Selector cho đường dẫn vào trang thông tin chi tiết



Hình 3.5: Selector cho đường dẫn vào trang kế tiếp

Để xác định được URL đến các trang đích, cần phải xác định được vị trí của thuộc tính href của thẻ <a> chứa các link đó thông qua XPATH Selector và CSS Selector trên DOM. Cụ thể được

thể hiện trong bảng sau:

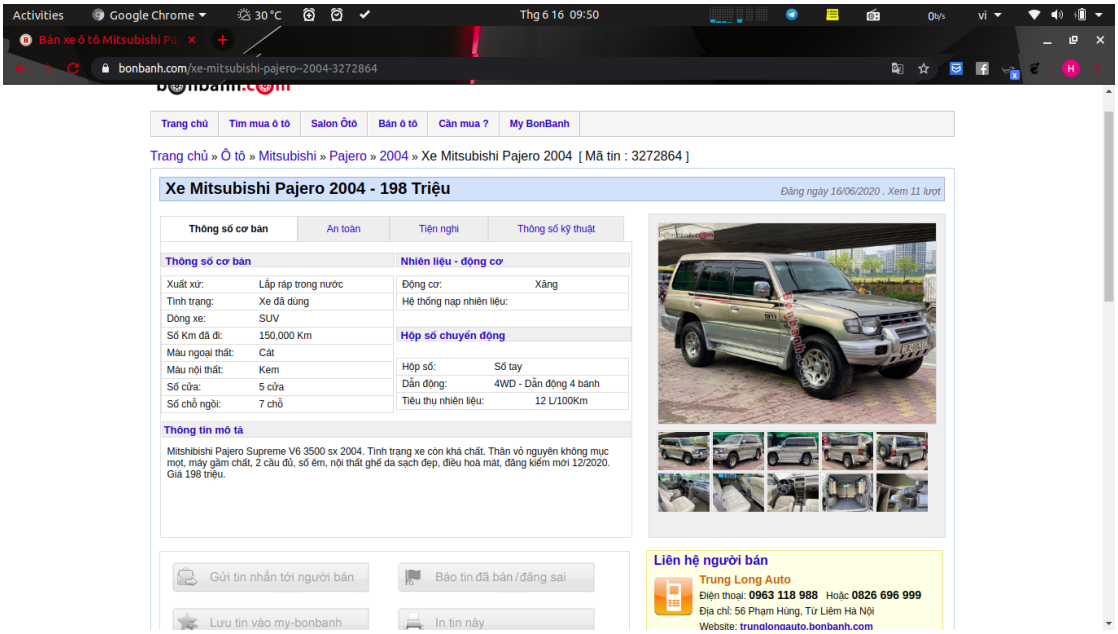
	Loại Selector	Giá trị
Chi tiết sản phẩm	CSS Selector	...
Trang kế tiếp	CSS Selector	...

Bảng 3.1: Tập luật khai phá của trình thu thập dữ liệu

3.1.3 Xây dựng luật trích chọn dữ liệu

Trang thông tin chi tiết sản phẩm có 3 mục chính để thu thập dữ liệu:

- Phần điều hướng chứa các thông tin về hãng xe, serie xe, đời xe.
- Phần bảng chi tiết các thuộc tính kỹ thuật của xe.
- Mô tả sản phẩm.



Hình 3.6: Tập luật thu thập thông tin sản phẩm

Thu thập dữ liệu phần điều hướng

Phần điều hướng chứa các thông tin về đời xe, hãng xe, dòng xe. Em sử dụng tập luật sử dụng Selector để trích chọn các thông tin trên từ trang HTML.

	Loại Selector	Giá trị
Hãng xe	CSS Selector	...
Dòng xe	CSS Selector	...
Đời xe	CSS Selector	...

Bảng 3.2: Tập luật trích trọn dữ liệu thông tin cơ bản

Thu thập dữ liệu phần bảng thông tin kỹ thuật

Bảng thông tin kỹ thuật gồm có các thông tin ở dạng thuộc tính - giá trị. Để thực thi thu thập thông tin này, em sử dụng XPATH Selector để xác định 1 danh sách có thứ tự toàn bộ các thuộc tính, và 1 danh sách có thứ tự toàn bộ các giá trị. Sau đó, em ghép 2 danh sách và lưu dưới dạng JSON.

	Loại Selector	Giá trị
Thuộc tính	XPATH Selector	...
Giá trị	XPATH Selector	...

Bảng 3.3: Tập luật trích chọn thông số kỹ thuật của xe

3.1.4 Kết quả xây dựng trình thu thập dữ liệu

Dữ liệu thu thập về được lưu dưới dạng JSON trong file items.json với cấu trúc như sau:

Thuộc tính	Mô tả	Ví dụ
abcd	ghik	...
abcd	ghik	...

Bảng 3.4: Schema của dữ liệu được thu thập về

3.2 Xử lý dữ liệu

3.2.1 Tiền xử lý dữ liệu

Dữ liệu được thu thập về và lưu trữ trong file items.json. Load dữ liệu này vào pandas DataFrame.

	url	brand	serie	model_year	price_vnd	price_dol	date	info	description
0	https://bonbanh.com/xe-toyota-vios-g-2018-3140449	Toyota	Vios	2018.0	515	22,079 USD	2020-03-23	{'Xuất xứ:': 'Lắp ráp trong nước', 'Tình trạng...	Xe đẹp. Đã sơn phủ gầm. Lót sàn da. Một số phụ...
1	https://bonbanh.com/xe-hyundai-accent-1.4-mt-2...	Hyundai	Accent	2020.0	465	19,936 USD	2020-03-23	{'Xuất xứ:': 'Lắp ráp trong nước', 'Tình trạng...	Quý khách vui lòng liên hệ vào Hotline để được...
2	https://bonbanh.com/xe-hyundai-i10-grand-1.0-m...	Hyundai	i10	2014.0	215	9,218 USD	2020-03-23	{'Xuất xứ:': 'Nhập khẩu', 'Tình trạng:': 'Xe đ...	Hyundai I10 2014 không khoan đục tư nhân chủ đ...
3	https://bonbanh.com/xe-mitsubishi-triton-4x2-m...	Mitsubishi	Triton	2019.0	555	23,794 USD	2020-03-23	{'Xuất xứ:': 'Nhập khẩu', 'Tình trạng:': 'Xe m...	Mitsubishi Triton 4x2 MT 2019 được trang bị Hi...
4	https://bonbanh.com/xe-mazda-3-1.5-at-2018-310...	Mazda	3	2018.0	639	27,395 USD	2020-03-23	{'Xuất xứ:': 'Lắp ráp trong nước', 'Tình trạng...	Bán xe Mazda 3 Sedan 1.5AT 2018, Facelift / Tr...

Dữ liệu thu thập về có 39211 dòng, mỗi dòng là thông tin của một tin rao bán xe. Trong khi các trường khác đều đầy đủ giá trị, trường năm sản xuất xe(model_year) bị thiếu và chỉ có 38613 dòng có giá trị. Em loại bỏ các bản ghi bị thiếu giá trị này.

Trường info là trường ở dạng JSON dưới dạng key-value, lưu lại thông tin được thu thập từ bảng thông số kĩ thuật của xe. Trong mỗi dòng dữ liệu, trường info chứa nhiều giá trị người dùng bỏ trống. Em load dữ liệu này vào một pandas DataFrame, và loại bỏ các trường bị thiếu giá trị. Kết quả các trường còn giữ lại như sau: Xuất xứ, Tình trạng, Dòng xe, Số Km đã đi, Màu ngoại thất, Màu nội thất, Số cửa, Số chỗ ngồi, Hộp số, Dẫn động. Sau đó, ghép các trường đã làm phẳng trong trường info với dữ liệu cũ, được một dataframe mới như sau.

	brand	serie	model_year	price_vnd	date	origin	status	category	used_dist	color_out	color_in	num_door	num_seat	gear	wheel_drive
0	Toyota	Vios	28	515	2020-03-23	Lắp ráp trong nước	Xe đã dùng	Sedan	45000	Cát	Kem	0	5	Số tự động	-
1	Hyundai	Accent	30	465	2020-03-23	Lắp ráp trong nước	Xe mới	Sedan	0	Đỏ	Kem	4	5	Số tay	FWD - Dẫn động cầu trước
2	Hyundai	i10	24	215	2020-03-23	Nhập khẩu	Xe đã dùng	Hatchback	0	Bạc	Nhiều màu	5	5	Số tay	FWD - Dẫn động cầu trước
3	Mitsubishi	Triton	29	555	2020-03-23	Nhập khẩu	Xe mới	Bán tải / Pickup	0	Trắng	Đen	4	5	Số tay	RFD - Dẫn động cầu sau
4	Mazda	3	28	639	2020-03-23	Lắp ráp trong nước	Xe đã dùng	Sedan	20000	Trắng	Đen	4	5	Số tự động	FWD - Dẫn động cầu trước

Làm sạch dữ liệu và điền giá trị

Để loại bỏ một số dòng dữ liệu sai sót do trình thu thập dữ liệu hoặc do sai sót của người đăng tin, em thực hiện một số bước làm sạch dữ liệu và điền đầy giá trị theo các luật sau:

- Loại bỏ các xe có status là xe mới.
- Các dòng dữ liệu có một trong các thông tin(wheel_drive, gear, num_door, num_seat, brand, serie, color_in, color_out) thuộc nhóm có tổng số phần tử nhỏ hơn 20 thì loại bỏ.
- Một số xe đã dùng có số km đã đi bằng 0 là các dòng dữ liệu người dùng không nhập. Thông tin đó được điền bằng trung bình số km đã đi của xe cùng dòng đó trong cùng năm.

Sau khi lọc DataFrame được lưu lại để phục vụ trực quan dữ liệu để tìm quy luật ẩn và loại nhiễu.

3.2.2 Trực quan hóa dữ liệu và loại nhiễu

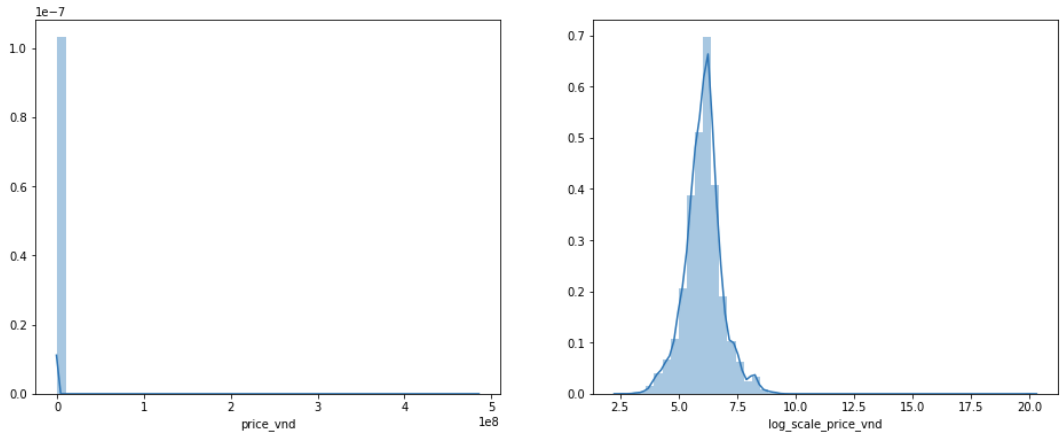
Dữ liệu người dùng trực tiếp nhập trên web để rao bán xe nên không thể tránh khỏi sai sót, nhầm lẫn. Trong phần này, em sẽ tiến hành trực quan hóa dữ liệu nhằm tìm quy luật và loại bỏ nhiễu ra khỏi bộ dữ liệu.

Dữ liệu giá xe

Giá xe là mục tiêu dự đoán của mô hình, nên đầu tiên, em sẽ phân tích trường thông tin này. Dưới đây là một số đặc tính thống kê của trường giá xe:

- Xe có giá thấp nhất là 13 triệu.
- Xe có giá cao nhất là 485 tỷ.
- 25% số xe có giá dưới 275 triệu.

- 50% số xe có giá dưới 448 triệu.
- 75% số xe có giá dưới 650 triệu.
- Giá xe trung bình là 19 tỷ 371 triệu.
- Độ lệch chuẩn của giá xe là 301 tỷ.

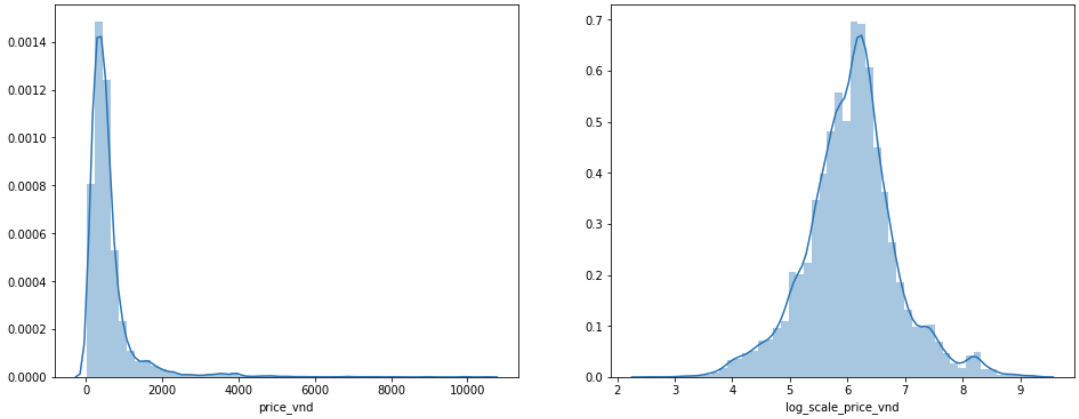


Hình 3.7: Phân bố giá xe khi chưa lọc nhiễu

Nhận thấy, dữ liệu về giá xe rất bất thường, các nhiễu dữ liệu do người dùng nhập sai. Ví dụ như xe cũ giá 485 tỷ là không hợp lý, do lỗi sai. Sau khi kiểm tra kỹ lại bộ dữ liệu và đối chiếu với thông tin đã sửa trên website người dùng đã sửa lại, em thống kê và sửa lỗi các lỗi người dùng nhập sai về giá như sau:

id	Dòng xe	Loại xe	Giá sai	Giá đúng	Ghi chú
66	Morning	Hatchback	26 triệu	260 triệu	
2713	Vios	Sedan	32 triệu	320 triệu	
20740	Mazda 3	Sedan	1 tỷ 700 triệu	Bỏ	Xe đắt lên vì biển số đẹp, là 1 lý do bất thường.
870	Tucson	SUV	485 tỷ	485 triệu	
31429	Xpander	SUV	1 tỷ 400 triệu	Bỏ	Xe đắt lên vì biển số đẹp, là 1 lý do bất thường.
3542	Jolie	SUV	145 tỷ	145 triệu	

Bảng 3.5: Mô tả một số lỗi sai gây nhiễu dữ liệu do người dùng nhập sai

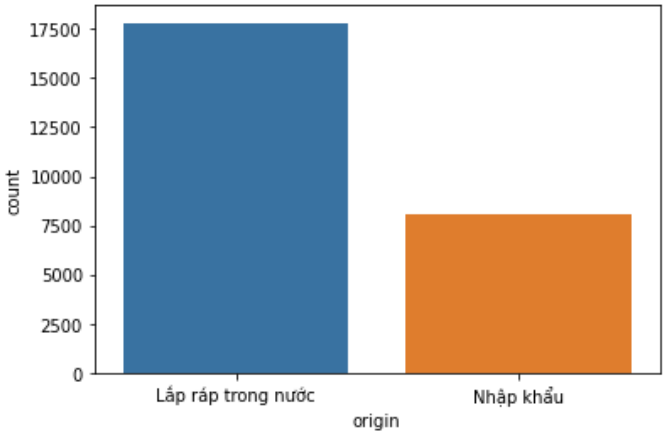


Hình 3.8: Phân bố giá xe sau khi đã lọc nhiễu

Sau khi sửa một số lỗi sai của người dùng nhập sai giá, phân bố giá xe đều và mịn hơn rất nhiều. Giá xe tối đa chỉ khoảng 10 tỷ, thuộc về dòng xe sang đời mới. Phân bố giá xe với log-scale gần đạt phân phối Gauss.

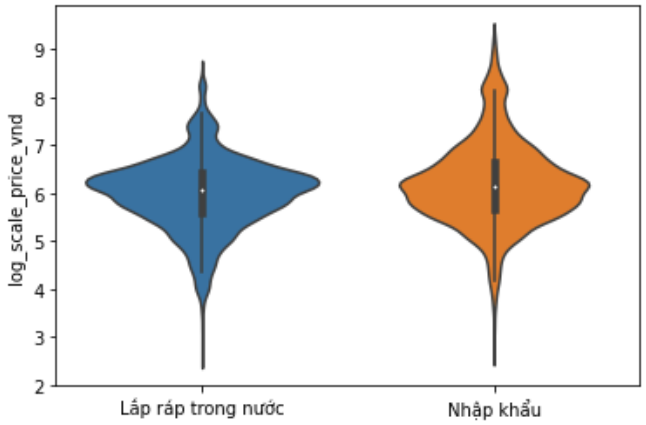
Dữ liệu xuất xứ xe

Trong hơn 25000 xe cũ được rao bán, trong đó có gần hơn 17500 xe được lắp ráp trong nước, còn lại khoảng 7500 xe nhập khẩu.



Hình 3.9: Đồ thị số lượng xe theo xuất xứ

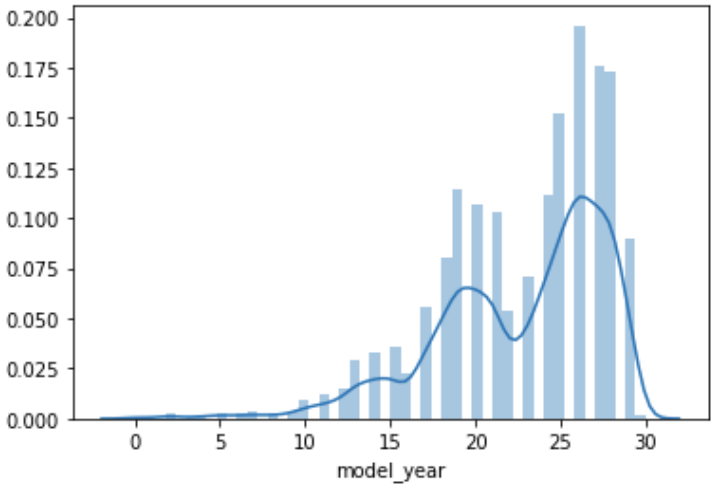
So sánh giá của các xe nhập khẩu và xe lắp ráp trong nước, nhận thấy rằng giá trung bình của 2 loại xe này không quá khác biệt, tuy vậy, số xe nhập khẩu ở dòng xe đắt tiền chiếm đại đa số.



Hình 3.10: Phân bố giá xe theo xuất xứ

Dữ liệu năm sản xuất xe

Với quy ước năm 1990 là 0, năm 2020 là 30, dữ liệu về số xe theo năm sản xuất được thể hiện trong hình sau



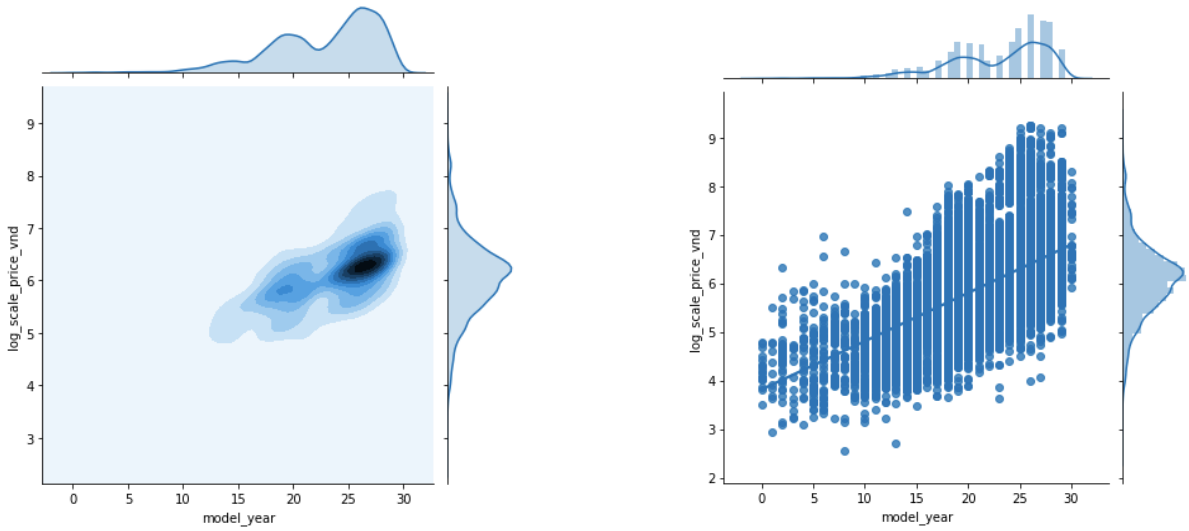
Hình 3.11: Phân bố số lượng xe theo năm sản xuất

Nhận thấy, chủ yếu các xe rao bán được sản xuất từ năm 2010 trở về sau chiếm đa số với các thông số thống kê như sau:

- 25% số xe được sản xuất trước năm 2009
- 50% số xe được sản xuất trước năm 2014
- 75% số xe được sản xuất trước năm 2017

Dưới đây là phân tích về giá xe cũ theo năm sản xuất

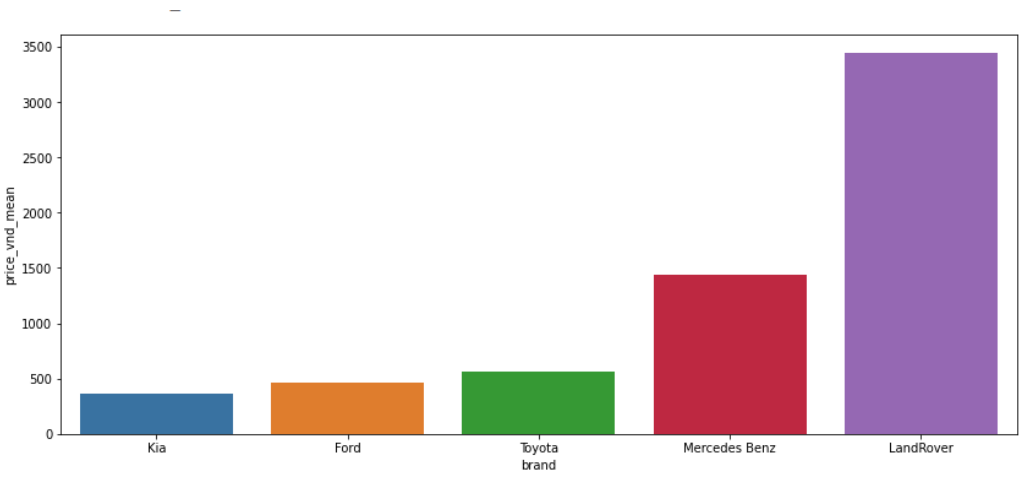
Giá xe tăng tỉ lệ thuận với năm sản xuất, hiển nhiên vì xe càng mới càng đắt. Tuy vậy, quan sát kĩ thì vẫn thấy một vài điểm dữ liệu bất thường như ở hình bên phải.



Hình 3.12: Phân bố giá xe theo năm sản xuất

Dữ liệu về hãng xe

Khảo sát giá xe theo các hãng tiêu biểu:



Hình 3.13: Phân bố giá xe theo hãng sản xuất

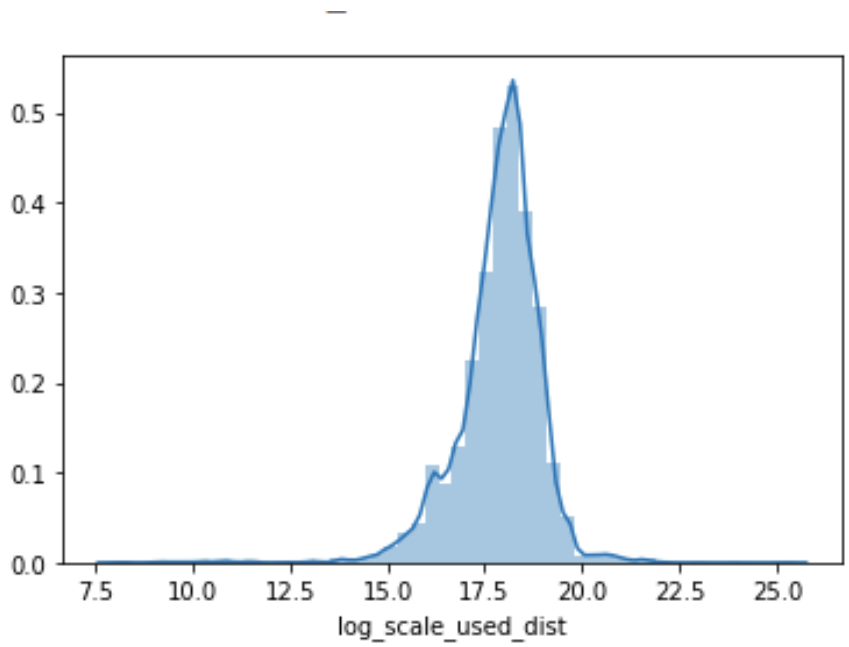
Chi tiết thống kê về giá xe của một số hãng tiêu biểu trong bảng sau

Hãng	Số mẫu	Trung bình	Độ lệch chuẩn
Daewoo	983	145.07	104.69
Kia	3745	360.94	199.35
Ford	2531	467.22	255.92
Toyota	6418	565.78	465.59
Mercedes Benz	1932	1436.31	983.12
Lexus	462	2674.52	1744.6
LandRover	237	3441.92	2268.11

Bảng 3.6: Thống kê giá xe theo hãng sản xuất

Dữ liệu số km đã đi

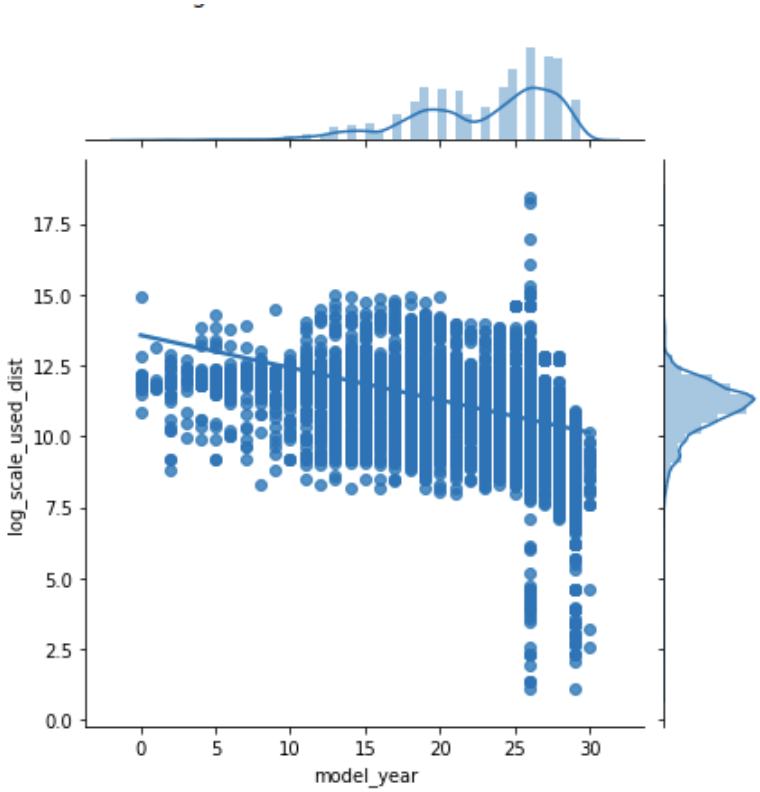
Dữ liệu về số km đã đi của xe được thể hiện như sau



Hình 3.14: Phân bố dữ liệu số km đã đi

Một số thống kê về số km đã đi(đơn vị loga):

- Giá trị min là 8, max là 25.3284
- 25% số xe có số km đã đi nhỏ hơn 17,4
- 50% số xe có số km đã đi nhỏ hơn 18.0202
- 75% số xe có số km đã đi nhỏ hơn 18.4904
- Giá trị trung bình là 17.8763, độ lệch chuẩn là 1.0680, phân phối có dạng phân phối Gauss



Hình 3.15: Phân bố giá xe theo số km đã đi

Thống kê số km đã đi được(đơn vị loga) theo năm sản xuất.

Nhận thấy xuất hiện nhiều nhiễu trong dữ liệu, qua quan sát đồ thị trên, em tiến hành lọc bỏ các nhiễu theo luật sau:

- Các xe sản xuất trước năm 2010, có số km đã đi dưới 7.5 hoặc trên 15 thì loại bỏ.
- Các xe sản xuất sau năm 2010, trước năm 2015, có số km đã đi dưới 7.5 hoặc trên 14 thì loại bỏ.
- Các xe sản xuất sau năm 2015, trước năm 2018, có số km đã đi dưới 7 hoặc trên 14 thì loại bỏ.
- Các xe sản xuất sau năm 2018, có số km đã đi dưới 7.5 hoặc trên 12 thì loại bỏ.

3.2.3 Hậu xử lý dữ liệu

Sau khi lọc nhiễu, dữ liệu sẽ được chuyển đổi qua các pipeline xử lý để vector hóa trước khi đưa vào mô hình học máy. Cụ thể:

- Các cột dữ liệu ở dạng Categorial như hãng xe, dòng xe, loại xe, ... sẽ được mã hóa dưới dạng One Hot Encoder.
- Cột số km đã đi sẽ biến đổi qua pipeline tuần tự như sau: lấy giá trị loga, standard scaler để đưa về phân phối có trung bình là 0
- Cột năm sản xuất có giá trị từ 0-30 nên sẽ chia cho 30 để đưa về đoạn [0, 1]

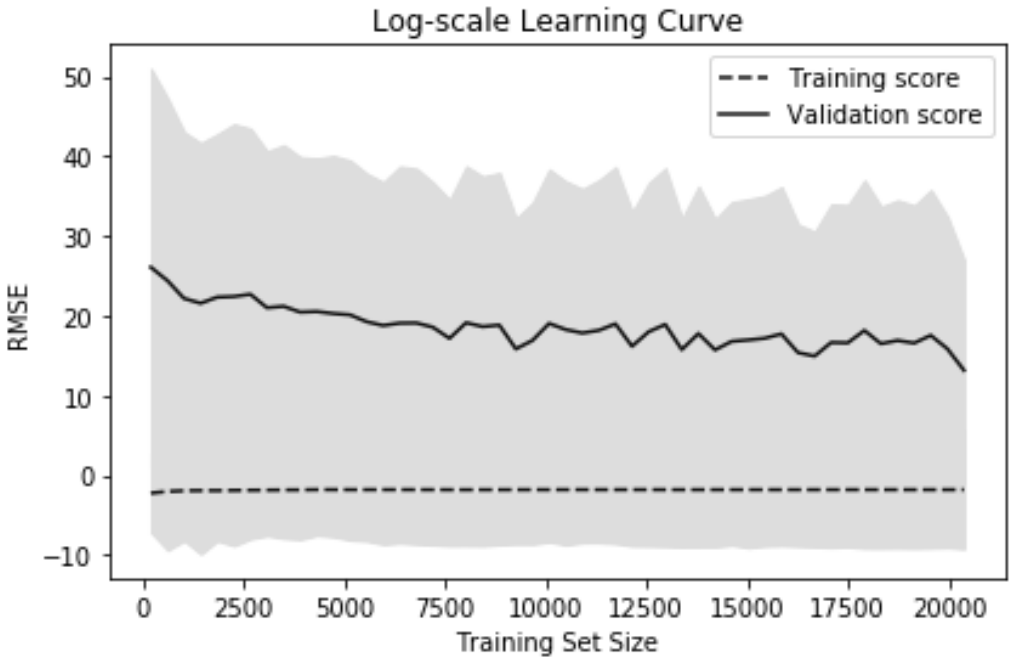
Sau quá trình này mỗi dòng dữ liệu được đưa về một vector để đưa vào mô hình học máy.

3.3 Xây dựng, đánh giá, lựa chọn mô hình học máy

Dữ liệu sau khi lọc nhiễu gồm có 25482 dòng dữ liệu, được chia làm 2 tập huấn luyện và tập kiểm thử theo tỉ lệ 9:1.

3.3.1 Mô hình Linear Regression

Sử dụng thư viện mô hình Linear Regression được cung cấp trong thư viện sklearn, huấn luyện trên training set, sử dụng k-fold validation và kiểm thử có kết quả như sau:



Hình 3.16: Learning curve giải thuật Linear Regression

Từ đồ thị trên có thể nhận thấy, thuật toán Linear Regression không đủ mạnh để mô tả dữ liệu. Kết quả trên tập test cũng cho thấy thuật toán không mô tả được quy luật trong dữ liệu với các chỉ số như sau:

- RMSE: 161.88
- MAE: 69.62
- MAPE: 11,56 %

3.3.2 Mô hình Support Vector Regression

Sử dụng Grid Search CV để tối ưu siêu tham số và sử dụng thuật toán Support Vector Regression của thư viện sklearn với k-fold validation với các tham số như sau:

- C: [1, 10, 30, 50, 100]
- epsilon: [0.0001, 0.001, 0.01]

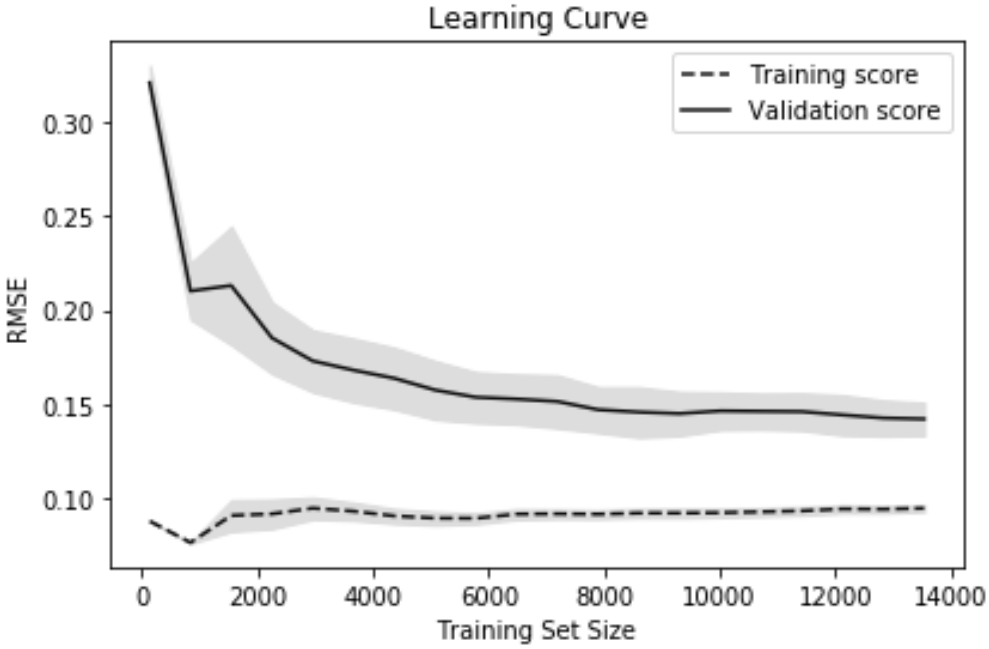
- gamma: [0.001, 0.01, 0.1]

Trong đó:

- C: Tham số regularization. Độ lớn của regularization tỉ lệ nghịch với C
- gamma: Hệ số cho kernel 'rbf'. gamma càng lớn, mô hình càng fit vào dữ liệu huấn luyện.
- epsilon: Hệ số chống lỗi cho nhiễu. epsilon càng nhỏ, mô hình càng fit vào dữ liệu huấn luyện, nhưng sẽ gây overfitting trên tập test.

Sau khi tìm kiếm trong bộ siêu tham số, mô hình Support Vector Regression tốt nhất với các siêu tham số như sau:

- C: 30
- epsilon: 0.01
- gamma: 0.01



Hình 3.17: Learning curve giải thuật SVR

Kết quả đánh giá mô hình trên tập kiểm thử như sau:

- RMSE: 123.4
- MAE: 52.66
- MAPE: 8,31 %

3.3.3 Mô hình Decision Tree Regression

Cài đặt sử dụng thuật toán Decision Tree Regression và Grid Search CV để tối ưu siêu tham số của mô hình với các siêu tham số sau:

- criterion :['mse','mae', 'friedman_mse']
- max_depth: [50, 80, 100, 300, 500, 1000]
- min_samples_split: [2, 5, 10]
- min_samples_leaf: [2, 5, 10]
- max_features: ['auto', 'log2', 'sqrt']

Trong đó:

- criterion: Tiêu chí đánh giá lỗi của mô hình khi huấn luyện.
- max_depth: Chiều sâu tối đa cho cây, cây có độ sâu càng lớn càng fit dữ liệu huấn luyện tốt, nhưng có thể gây ra overfitting. Ngược lại, nếu độ sâu quá thấp thì không thể fit dữ liệu gây ra underfitting.
- min_samples_split: Số mẫu tối thiểu cần thiết để phân chia một node trong.
- min_samples_leaf: Số mẫu tối thiểu cần thiết để là một node lá
- max_features: số feature tối đa được xét khi phân chia 1 node.

Sau khi tìm kiếm trong bộ siêu tham số, mô hình Decision Tree Regression tốt nhất với các siêu tham số như sau:

- criterion: friedman_mse
- max_depth: 1000
- min_samples_split: 10
- min_samples_leaf: 2
- max_features: auto

Kiểm định trên tập test thu được kết quả như sau:

- RMSE: 117.51
- MAE: 30.7
- MAPE: 5.1%

3.3.4 Mô hình Random Forest Regression

Cài đặt sử dụng thuật toán Random Forest Regression và Grid Search CV để tối ưu siêu tham số của mô hình với các siêu tham số sau:

- criterion:['mse']
- max_depth: [20, 30, 40, 50, 80, 100]
- min_samples_split: [2, 5, 10]
- min_samples_leaf: [1, 2, 4, 8],
- n_estimators: [50, 100, 150, 200, 300, 500, 1000]

Trong đó:

- n_estimators: số cây trong rừng
- criterion: Tiêu chí đánh giá lỗi của mô hình khi huấn luyện.
- max_depth: Chiều sâu tối đa cho cây trong rừng.
- min_samples_split: Số mẫu tối thiểu cần thiết để phân chia một node trong.
- min_samples_leaf: Số mẫu tối thiểu cần thiết để là một node lá

Sau khi tìm kiếm trong bộ siêu tham số, mô hình Random Forest Regression tốt nhất với các siêu tham số như sau:

- criterion: mse
- max_depth: 50
- min_samples_split: 10
- min_samples_leaf: 1
- n_estimators: 300

Kiểm định trên tập test thu được kết quả như sau:

- RMSE: 39.08
- MAE: 21.4
- MAPE: 4.28%

3.3.5 Đánh giá, tổng hợp kết quả, lựa chọn mô hình

Thông số	Linear Regression	Support Vector Regression	Decision Tree Regression	Random Forest Regression
RMSE	161.88	123.4	117.51	39.08
MAE	69.62	52.66	30.7	21.4
MAPE	11,56	8,31	5.1	4.28

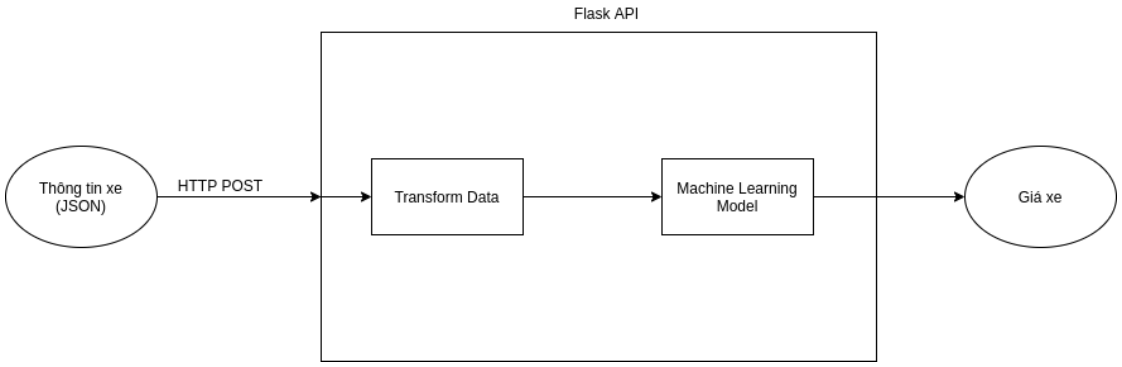
Bảng 3.7: Tổng hợp kết quả của các mô hình Học Máy

Qua đó có thể thấy mô hình Decision Tree và Random Forest có độ chính xác cao so với các mô hình khác với MAPE là 5.1% và 4.28% tương ứng. Mô hình Random Forest có độ chính xác tốt hơn, tuy vậy, do cần huấn luyện 300 Decision Tree rồi lấy kết quả trung bình nên thời gian huấn luyện lâu hơn. Tuy vậy, với tài nguyên tính toán hiện tại, chúng ta có thể đánh đổi thời gian, chi phí tính toán lấy độ chính xác. Như vậy, em sẽ lựa chọn mô hình Random Forest để triển khai.

3.4 Triển khai hệ thống định giá tự động

Hệ thống định giá tự động mà em xây dựng trong đồ án này cho phép định giá xe ô tô cũ qua các thông tin về xe thông qua HTTP Request đến API của hệ thống. Hệ thống là một hệ thống con chỉ chịu trách nhiệm đánh giá giá của tài sản, đầu vào của hệ thống này được cung cấp bởi hệ thống lớn của doanh nghiệp.

3.4.1 Sơ đồ kiến trúc tổng quan



Hình 3.18: Sơ đồ kiến trúc hệ thống định giá tự động

Đầu vào của hệ thống là thông tin xe dưới dạng JSON, được đính kèm trong body của HTTP Request. Hệ thống sẽ chuyển đổi dữ liệu đầu vào thành feature vector qua pipeline transform data. Sau đó, mô hình Machine Learning đã xây dựng sẽ dự đoán trên feature vector đó, đưa ra giá xe được dự đoán.

Thông tin	Loại dữ liệu	Mô tả	Giá trị ví dụ
brand	category	Hãng xe	Mazda
serie	category	Dòng xe	Cx5
model_year	number	Năm sản xuất	27
origin	category	Nguồn gốc xe	Nhập khẩu
category	category	Loại xe	SUV
used_dist	number	Số km đã đi	54000
color_out	categorical	Màu sơn ngoài	Xanh
color_in	categorical	Màu nội thất	Đen
num_door	categorical	Số cửa	5
num_seat	categorical	Số ghế	5
gear	categorical	Loại số	Số tự động
wheel_drive	categorical	Loại dẫn động	FWD - Dẫn động cầu trước

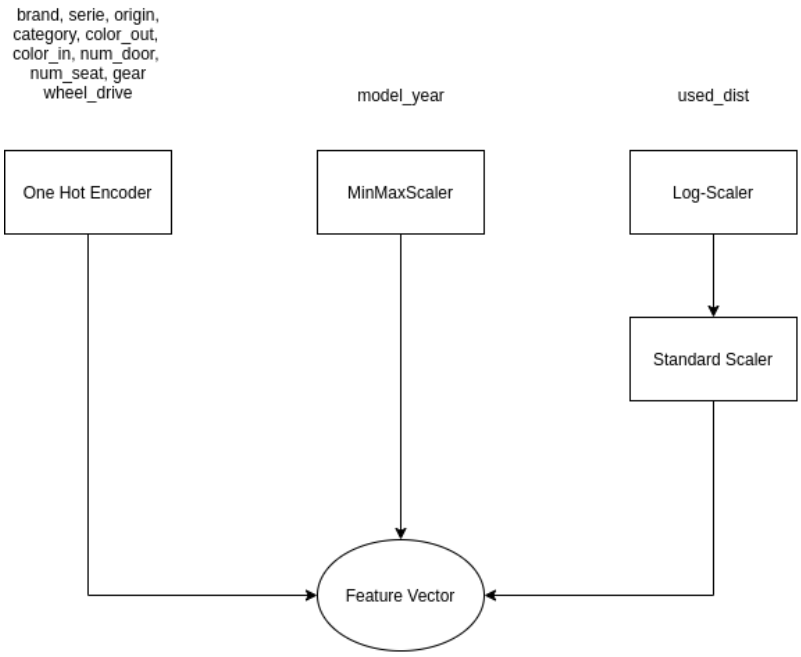
Bảng 3.8: Các thông tin đầu vào hệ thống

3.4.2 Xây dựng pipeline transform data

Dữ liệu đầu vào được chia làm 2 nhóm:

- Dữ liệu dạng categorical: brand, serie, origin, category, color_out, color_in, num_door, num_seat, gear, wheel_drive
- Dữ liệu dạng numerical: model_year, used_dist

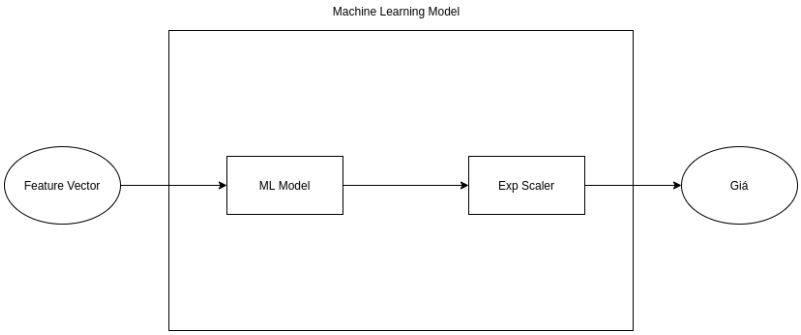
Trong đó, nhóm dữ liệu dạng categorical sẽ được encode bằng phương pháp One Hot Encoder, nhóm dữ liệu dạng numerical sẽ được encode qua các pipeline để scale và normalize dữ liệu. Sau quá trình encode dữ liệu sẽ thu được vector đặc trưng để đưa vào mô hình học máy để đưa ra dự đoán.



Hình 3.19: Sơ đồ khối pipeline transform data

3.4.3 Triển khai mô hình học máy

Dựa vào mô hình học máy đã lựa chọn ở phần 3.3, em lưu lại và triển khai API web services với thư viện Flask. Đầu vào của phần này là Vector đặc trưng đã thu được ở phần trên. Do ở phần xử lý dữ liệu, đầu ra của thuật toán là giá ở thang loga, nên sau đó cần phải đưa về giá trị ở thang đo thông thường qua hàm mũ.



Hình 3.20: Sơ đồ khối triển khai mô hình Học máy

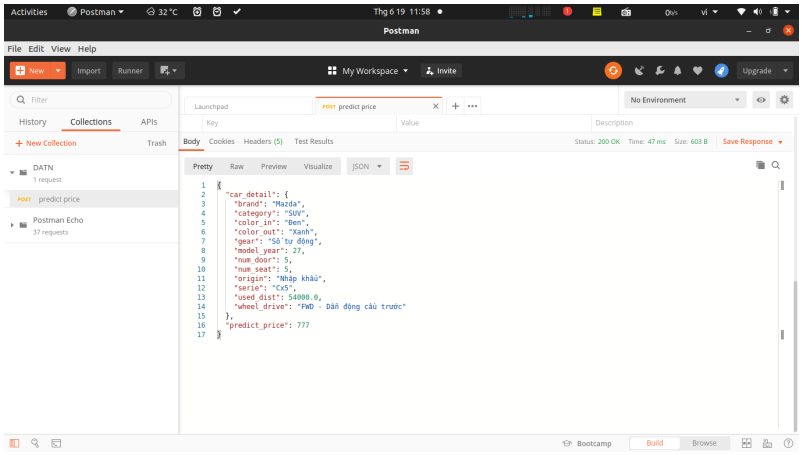
3.4.4 Kết quả triển khai hệ thống

Hệ thống được triển khai dưới dạng JSON Web Services, em sử dụng phần mềm POSTMAN để kiểm tra với bộ đầu vào như sau:

- brand: Mazda
- serie: Cx5
- model_year: 27
- origin: Nhập khẩu

- category: SUV
- used_dist: 54000
- color_out: Xanh
- color_in: Đen
- num_door: 5
- num_seat: 5
- gear: Số tự động
- wheel_drive: FWD - Dẫn động cầu trước

Kết quả, hệ thống trả về giá xe là 777 triệu như trong hình bên dưới.



Hình 3.21: Kết quả demo hệ thống

KẾT LUẬN

Đồ án đã giải quyết được bài toán định giá tài sản tự động cho doanh nghiệp tài chính ứng dụng Học máy. Các mục tiêu đề ra ban đầu của đồ án đã cơ bản hoàn thành. Cụ thể như sau:

- Đã xây dựng được hệ thống định giá tài sản cho xe ô tô cũ với mức sai số 4.28%.
- Hệ thống hoạt động ổn định, cung cấp qua dịch vụ JSON Web API.
- Tuy nhiên độ trễ xử lý của hệ thống khi dự đoán là 25ms, chưa đạt được mục tiêu 20ms như đã đề ra ban đầu.

Dưới sự hướng dẫn tận tâm của thầy, em đã cơ bản hoàn thành được các gói công việc đã đề ra. Cụ thể:

- Xây dựng trình thu thập dữ liệu Web.
- Xử lý dữ liệu sau khi thu thập, trực quan hóa và loại nhiễu.
- Xây dựng mô hình Học máy.
- Triển khai ứng dụng trên nền tảng JSON Web Services.

Đồ án được thực hiện dựa trên dữ liệu được thu thập từ trên website, do vậy chất lượng của dữ liệu chưa hẳn đảm bảo, có chứa nhiều sai sót, nhiễu mang tính hệ thống khi người dùng tăng giá bán có chủ đích hay việc họ vô tình nhập sai. Dữ liệu là phần mang tính quyết định lớn trong sự thành công của các mô hình học máy. Khi nguồn dữ liệu không đủ chất lượng, mô hình học máy có thể có những sai sót mang tính thiên lệch theo bộ dữ liệu huấn luyện của nó.

Mặt khác, đồ án mới chỉ giải quyết một phần nhỏ trong nghiệp vụ định giá tài sản tự động, cụ thể là định giá xe ô tô cũ. Nghiệp vụ của doanh nghiệp tài chính, ngân hàng còn nhiều mục tài sản cần định giá khác như nhà đất. Để có thể giải quyết trọn vẹn vấn đề định giá tài sản tự động, hệ thống cần phải mở rộng ra để có thể định giá nhiều loại tài sản phát sinh trong hoạt động của các doanh nghiệp này.

Hướng phát triển của đồ án
Trong tương lai, đồ án này có thể được phát triển nhằm mở rộng tính năng và hiệu năng như:

- Tối ưu lại pipeline xử lý dữ liệu và mô hình học máy, để giảm độ trễ xử lý của hệ thống.
- Thu thập nhiều hơn dữ liệu về số lượng, cũng như tăng chất lượng của dữ liệu trước khi đưa vào huấn luyện mô hình Học Máy.
- Mở rộng tính năng của đồ án, xây dựng hệ thống định giá cho nhiều loại tài sản để phù hợp với yêu cầu của doanh nghiệp.

TÀI LIỆU THAM KHẢO

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

RE Fan, PH Chen, and Chih-Jen Lin. Working set selection using second order information for training svm. *Journal of Machine Learning Research*, 6:1889–1918, 01 2005.

Aurélien Géron. Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligentsystems. Paperback, April 2017.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1:14 – 23, 01 2011.

Wes McKinney. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 1 edition, February 2013.

Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.