**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

-------------------

A close up of a sign

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỌC: NHẬP MÔN KHOA HỌC DỮ LIỆU**

ĐỀ TÀI

**Dự đoán giá nhà đất**

|  |
| --- |
| PGS. TS. Thân Quang Khoát |
| TS. Trần Việt Trung |
| TS. Nguyễn Kiêm Hiếu |
| TS. Nguyễn Thị Oanh |
| TS. Bùi Thị Mai Anh |

**GVHD:**

|  |  |
| --- | --- |
| Thành viên nhóm | MSSV |
| Nguyễn Tiến Anh | 20172953 |
| Nguyễn Văn Hùng | 20173152 |
| Nguyễn Văn Tiến | 20173402 |
| Nguyễn Phú Vượng | 20173473 |

**Nhóm 2:**

***Hà Nội, ngày 10 tháng 12 năm 2020***

MỤC LỤC

[LỜI MỞ ĐẦU 4](#_Toc58923856)

[1. Bài toán dự đoán giá nhà đất 5](#_Toc58923857)

[1.1 Tổng quan về bài toán dự đoán giá nhà đất 5](#_Toc58923858)

[1.2 Phương pháp giải quyết bài toán dự đoán giá nhà đất 5](#_Toc58923859)

[2. Công nghệ sử dụng 5](#_Toc58923860)

[2.1 Tìm kiếm và thu thập dữ liệu thô từ Internet 5](#_Toc58923861)

[2.2 Lưu trữ và xử lý dữ liệu thô thành dữ liệu “sạch” trên hệ thống phân tán 6](#_Toc58923862)

[2.2.1 Tìm hiểu về hadoop distributed file system (hdfs) 6](#_Toc58923863)

[2.2.2 Tìm hiểu về spark 8](#_Toc58923864)

[2.3 Áp dụng giải thuật học máy dự đoán giá nhà 9](#_Toc58923865)

[2.3.1 Giải thuật hồi quy tuyến tính 9](#_Toc58923866)

[2.3.2 Giải thuật k láng giềng gần nhất 9](#_Toc58923867)

[3. Xử lý dữ liệu 13](#_Toc58923868)

[3.1 Dữ liệu và xử lý dữ liệu 13](#_Toc58923869)

[3.1.1 Dữ liệu thô 13](#_Toc58923870)

[3.1.2 Xử lý dữ liệu 14](#_Toc58923871)

[3.1.3 Khám phá dữ liệu 14](#_Toc58923872)

[3.2 Phương pháp đánh giá giải thuật 21](#_Toc58923873)

[3.2.1 Mean Square Error 21](#_Toc58923874)

[3.2.2 R2 Score 21](#_Toc58923875)

[3.3 Thực nghiệm 21](#_Toc58923876)

[3.3.1 Hồi quy tuyến tính 21](#_Toc58923877)

[3.3.2 K láng giềng gần nhất 22](#_Toc58923878)

[3.3.3 Nhận xét 23](#_Toc58923879)

[4. Kết luận 23](#_Toc58923880)

[4.1 Kết luận chung 23](#_Toc58923881)

[4.2 Những khó khăn gặp phải 24](#_Toc58923882)

[4.3 Định hướng phát triển 24](#_Toc58923883)

[5. Tài liệu tham khảo 24](#_Toc58923884)

LỜI MỞ ĐẦU

Ngày nay, sự phát triển của công nghệ thông tin đã xâm nhập vào hầu hết các mặt của đời sống xã hội, với những ứng dụng rộng rãi hỗ trợ cho con người trên nhiều lĩnh vực, trong đó có lĩnh vực bất động sản. Với sự phát triển vượt bậc của khoa học trong những năm gần đây, ta có thể xây dựng một hệ thống dự đoán giá nhà đất bằng những phương pháp của học máy.

Nhóm em xin gửi lời cảm ơn chân thành tới thầy Thân Quang Khoát, thầy Trần Việt Trung, thầy Nguyễn Kiêm Hiếu, cô Nguyễn Thị Oanh, cô Bùi Thị Mai Anh đã tin tưởng cho nhóm em chọn đề tài này, đồng thời đã nhiệt tình hướng dẫn nhóm em trong suốt học kì vừa qua, giúp nhóm em học được nhiều kiến thức mới và đặc biệt cho nhóm em cách nhìn khác hơn về các lĩnh vực trong ngành công nghệ thông tin.

Do nhiều nguyên nhân cả chủ quan và khách quan, chủ yếu là do kiến thức còn hạn hẹp và thời gian tìm hiểu không có nhiều, nghiên cứu chúng em thực hiện còn nhiều hạn chế, mong nhận được sự chỉ dẫn của các thầy, cô để nhóm em hoàn thiện hơn trong tương lai.

# bài toán dự đoán giá nhà đất

## Tổng quan về bài toán dự đoán giá nhà đất

Bài toán của nhóm em là dự đoán được giá cả ở thời điểm hiện tại của một căn nhà (nhà đất, có sổ đỏ chính chủ) ở hầu hết những huyện trên đất nước Việt Nam dựa vào một số đặc điểm nổi bật của căn nhà, giúp những người muốn bán nhà và những ai muốn mua nhà có thể tham khảo và đưa ra mức giá phù hợp hơn với căn nhà. Giúp tiết kiệm thời gian và công sức, chi phí tìm hiểu thị trường cho khách hàng.

## Phương pháp giải quyết bài toán dự đoán giá nhà đất

Quá trình giải quyết bài toán gồm các bước:

* Tìm kiếm và thu thập dữ liệu thô từ internet.
* Lưu trữ và xử lý dữ liệu thô thành dữ liệu “sạch” trên hệ thống phân tán.
* Áp dụng giải thuật học máy để dự đoán được giá nhà.

# Công nghệ sử dụng

## Tìm kiếm và thu thập dữ liệu thô từ Internet

Hiện nay, thông tin là nguồn tài nguyên quý giá, việc sở hữu được nhiều thông tin cung cấp cho ta rất nhiều lợi thế. Đặc biệt với sự bùng nổ của internet, việc tiếp cận với nguồn thông tin khổng lồ trở nên dễ dàng hơn, ngày càng nhiều thư viện, công cụ hỗ trợ thu thập nguồn dữ liệu đó, trong đó có thư viện scrapy.

***Tìm hiểu về Scrapy***

Scrapy là một framework mã nguồn mở, không chỉ là một thư viện. Scrapy quản lý các yêu cầu, chuyển đổi các trang web HTML, thu thập dữ liệu, và lưu dữ liệu đó ở định dạng mong muốn của người dùng. Vì thế mọi nhu cầu hầu như có thể xử lý được bằng scrapy mà không cần phải cài đặt nhiều thư viện riêng lẻ khác nhau. Scrapy cũng cho phép sử dụng middlewares để có thể thu thập được dữ liệu từ những trang web yêu cầu xác thực tài khoản người dùng.

***Các thành phần của framework scrapy***

*Scrapy Engine*: điều khiển luồng dữ liệu giữa tất cả các thành phần của hệ thống với nhau.

*Scheduler*: nhận các yêu cầu từ scrapy engine và lập lịch trình cho hàng đợi để cho quá trình tiếp theo.

*Downloader*: lấy các trang web và đưa chúng vào engine.

*Spiders*: chuyển đổi phản hồi và trích xuất các dữ liệu cần thiết cũng như các yêu cầu bổ sung.

*Item pipeline*: xử lý các dữ liệu sau khi được trích xuất bởi spiders.

*Downloader middlewares*: xử lý các yêu cầu khi chúng được chuyển giữa engine và downloader.

*Spider middlewares*: xử lý các đầu vào và đầu ra của quá trình spider.

***Tại sao lại sử dụng Scrapy?***

Scrapy là một framework mạnh mẽ, nhóm em sử dụng scrapy bởi một số lý do sau:

* Cho phép thu thập dữ liệu chỉ cần cung cấp đường dẫn.
* Có thể thu thập dữ liệu đồng thời giữa nhiều trang web.
* Tự động tìm đến file Robots.txt và thẻ <meta> giúp “văn minh” khi thu thập dữ liệu tại trang web mà không thuộc sở hữu của mình.
* Dựa trên sơ đồ cấu trúc XML của trang web để thu thập đúng dữ liệu cần thiết trong thời gian tối ưu.
* Lọc và loại bỏ những đường dẫn trùng lặp.
* Thử lại khi lỗi phát sinh trong quá trình thu thập dữ liệu.

## Lưu trữ và xử lý dữ liệu thô thành dữ liệu “sạch” trên hệ thống phân tán

Để lưu trữ được lượng dữ liệu lớn thì Hadoop distributed file system (HDFS) là một lựa chọn tốt, và xử lý lượng dữ liệu đó một cách phân tán không thể thiếu Spark.

### Tìm hiểu về hadoop distributed file system (hdfs)

Hadoop distributed file system (hay gọi tắt là HDFS) là một hệ thống tệp tin phân tán dựa trên ngôn ngữ Java, HDFS cho phép lưu trữ lượng dữ liệu lớn thông qua nhiều nút trong cụm Hadoop.

***Tính năng của HDFS***

*Cost*: sử dụng ít tài nguyên phần cứng hơn, nếu không dùng HDFS thì chi phí cho việc mở rộng hệ thống sẽ rất lớn. Hay nói cách khác, mở rộng nhiều nút hơn trong cụm Hadoop sẽ tiết kiệm hơn.

*Variety and Volumn of Data*: HDFS cho phép lưu trữ lượng lớn dữ liệu của nhiều loại dữ liệu khác nhau, có cấu trúc, không có cấu trúc hay bán cấu trúc.

*Reliability and Fault Tolerence*: Khi lưu trữ dữ liệu trên HDFS, dữ liệu được chia thánh từng khối và lưu trữ phân tán trên các máy trong cụm hadoop. Dữ liệu được sao chép thành nhiều bản, thường là 3, do đó khi một nút nào đó bị hỏng dữ liệu thì vẫn còn dữ liệu đó ở các nút khác thay thế.

*Data Integrity*: HDFS thường xuyên kiểm tra tính đúng đắn của dữ liệu được lưu trữ trên nó. Nếu HDFS thấy bất cứ lỗi nào, nó sẽ báo cho nút quản lý của dữ liệu đó. Và nút quản lý đó sẽ tạo bản sao mới và xoá dữ liệu lỗi đi.

*High Throughput*: Bởi vì khả năng lưu trữ và xử lý dữ liệu phân tán và song song của HDFS nên HDFS chỉ cần ít thời gian hơn so với các máy tính thông thường.

*Data Locallity*: với HDFS, chúng ta không cần phải gộp các phần dữ liệu từ các nút về một nơi trung tâm để xử lý mà có thể làm việc đó ngay trên các nút, giúp tiết kiệm chi phí khi di chuyển dữ liệu lớn

***Tại sao sử dụng HDFS?***

Nhóm em sử dụng HDFS để lưu trữ dữ liệu bởi các ưu điểm của HDFS:

*Lưu trữ phân tán*: khi truy cập HDFS từ bất kỳ các nút nào trong cụm Hadoop, người dùng sẽ cảm nhận như là truy cập vào một máy tính chứa dữ liệu duy nhất.

*Tính toán phân tán và song song*: bởi vì dữ liệu được phân chia ra nhiều nút trong cụm Hadoop, nên nếu không sử dụng HDFS, mà cần xử lý 10 TB dữ liệu mất 40 phút thì thực tế chỉ mất 4 phút nếu sử dụng HDFS gồm 10 nút.

*Khả năng mở rộng*: HDFS cho phép mở rộng hệ thống theo chiều ngang (thêm nhiều nút vào cụm hadoop đã có mà không cần phải dừng hệ thống lại, tiết kiệm được thời gian đã tính toán từ trước).

### Tìm hiểu về spark

Apache Spark (gọi tắt là Spark) là một công nghệ tính toán trên cụm, được thiết kế để cho việc tính toán nhanh. Spark dựa trên Hadoop MapReduce và nó mở rộng MapReduce để hiệu quả hơn cho nhiều loại tính toán khác nhau. Tính năng chính của Spark là khả năng tính toán trên cụm giúp tăng tốc độ xử lý của một phần mềm.

***Các thành phần của Spark***

*Apache Spark Core:* Spark Core là một engine thực thi chung, chạy nền cho tất cả các hàm khác được xây dựng trên nền tảng này. Cung cấp khả năng tính toán và tham chiếu dữ liệu ở các hệ thống lưu trữ bên ngoài.

*Spark SQL:* là một thành phần bên trên của Spark Core, đưa ra một kiểu dữ liệu trừu tượng gọi là SchemaRDD, cung cấp sự hỗ trợ cho dữ liệu có cấu trúc và bán cấu trúc.

*Spark Streaming:* thúc đẩy khả năng lập lịch nhanh của Spark Core để biểu diễn phân tích trực tuyến. Tiếp nhận dữ liệu theo từng tập nhỏ và biẻu diễn chuyển đổi RDD trên tập dữ liệu đó.

*MLlib:* là một framework học máy phân tán dựa trên Spark theo kiến trúc phân tán bộ nhớ của Spark. Spark MLlib nhanh gấp 9 lần so với phiên bản ổ cứng Hadoop của Apache Mahout.

*GraphX:* là một framework xử lý đồ thị phân tán dựa trên Spark. Cung cấp API cho việc biểu diễn tính toán đồ thị, có thể là đồ thị được người dùng định nghĩa sử dụng Pregel API.

***Tại sao lại sử dụng Spark?***

Nhóm em quyết định sử dụng Spark để xử lý dữ liệu bới vì các đặc điểm sau của Spark:

*Tốc độ:* Spark giúp chạy một ứng dụng trong cụm Hadoop, nhanh gấp 100 lần về phương diện bộ nhớ và nhanh gấp 10 lần khi chạy trên ổ đĩa. Điều này có thể giúp giảm số lượng các phép toán đọc ghi vào ổ đĩa. Spark lưu trữ dữ liệu đang được xử lý vào bộ nhớ ngay lập tức.

*Hỗ trợ nhiều ngôn ngữ*: Spark cung cấp API có sẵn viết bằng Java, Scala, Python. Vì vậy ta có thể viết ứng dụng bằng nhiều ngôn ngữ khác nhau.

*Phân tích nâng cao:* Spark không những hỗ trợ “Map” và “reduce” mà còn hỗ trợ truy vấn SQL, dòng dữ liệu, học máy, và các giải thuật đồ thị.

## Áp dụng giải thuật học máy dự đoán giá nhà

Giá nhà đất là một dạng số thực, nên không thể giải quyết bài toán bằng bài toán phân loại được. Nhóm em sử dụng cả hai giải thuật hồi quy tuyến tính và k láng giềng gần nhất để so sánh và chọn ra giải thuật tốt nhất.

### Giải thuật hồi quy tuyến tính

Giải thuật hồi quy tuyến tính cố gắng tính toán sự tương quan giữa dữ liệu đầu vào và kết quả tương ứng đầu ra. Ví dụ, sử dụng hồi quy tuyến tính để xét xem sự tương quan giữa các đặc điểm của căn nhà và giá nhà như thế nào từ các dữ liệu đầu vào, từ đó suy ra được một “công thức” gần đúng với các biến đầu vào là các đặc điểm của căn nhà, và đầu ra là giá của căn nhà đó. Từ đó chỉ cần biết được các đặc điểm của căn nhà, áp dụng “công thức” trên dự đoán được giá nhà là bao nhiêu.



Đầu vào của giải thuật hồi quy tuyến tính là một vectơ n chiều .

Đầu ra của giải thuật là một giá trị thực, được tính theo công thức đã xác định được từ quá trình đưa tập dữ liệu ban đầu vào giải thuật.

***Khi nào sử dụng giải thuật hồi quy tuyến tính***

Giải thuật hồi quy tuyến tính thực hiện các tính toán tuyến tính, đơn giản, dễ cài đặt và thuật toán dễ hiểu. Chi phí tính toán thấp (bởi vì các phép toán được thực hiện tuyến tính). Thời gian dự đoán nhanh, cho kết quả ngay, phù hợp với những dạng bài toán có độ phức tạp không quá cao và cần thời gian dự đoán ngay lập tức.

### Giải thuật k láng giềng gần nhất

Giải thuật k láng giềng gần nhất (K-Nearest Neighbor hay KNN) tìm ra k căn nhà (trong tổng số những căn nhà đã biết giá) có “khoảng cách” gần với căn nhà cần dự đoán giá nhất. Hay nói cách khác, từ những đặc điểm của các căn nhà đã biết giá và căn nhà cần dự đoán giá, thuật toán trên sẽ tính được k căn nhà có những đặc điểm giống với căn nhà cần dự đoán nhất, sau đó giá của căn nhà cần dự đoán sẽ được tính dựa trên giá của k căn nhà đó.

***Mô tả giải thuật kNN***

K láng giềng gần nhất (tên Tiếng Anh là K-Nearest Neighbor, viết tắt là KNN) là một giải thuật học máy có giám sát đơn giản, dễ cài đặt và có thể sử dụng để giải quyết cả hai dạng bài toán phân loại và hồi quy.

Ý tưởng của phương pháp học dựa trên láng giềng gần nhất đó là với một tập các ví dụ học, thuật toán đơn giản là lưu lại các ví dụ học, chưa xây dựng mô hình mô tả rõ ràng và tổng quát của hàm mục tiêu cần học. Đối với một ví dụ cần phân loại hay hồi quy, thuật toán xem xét ví dụ đó với các ví dụ học để gán giá trị của hàm mục tiêu (một nhãn hay một giá trị thực)

***Biểu diễn đầu vào của thuật toán***

Mỗi ví dụ x được biểu diễn là một vectơ n chiều trong không gian các vectơ X ∈ Rn.

x = (x1,x­1,…,xn), trong đó xi (∈R) là một số thực. Có thể áp dụng với hai kiểu bài toán: bài toán phân lớp (hàm mục tiêu có giá trị rời rạc, đầu ra của hệ thống là một trong số các giá trị rời rạc được xác định trước, là một trong các nhãn lớp) hoặc bài toán hồi quy (hàm mục tiêu có giá trị liên tục, đầu ra của hệ thống là một số thực không biết trước).

***Giải thuật phân lớp KNN***

Mỗi ví dụ học x được biểu diễn bởi 2 thành phần, mô tả của ví dụ: x=(x1,x2,…,xn), trong đó xi∈R, và giá trị đầu ra mong muốn: yx∈R (là một số thực). Trong giai đoạn học, thuật toán lưu lại các ví dụ học trong tập học D. Trong giai đoạn dự đoán, đối với mỗi ví dụ học x∈D, tính khoảng cách giữa x và z (ví dụ cần dự đoán), từ đó xác định được các láng giềng gần nhất của z, gồm k ví dụ học trong D gần nhất với z tính theo một hàm khoảng cách d.

***Các hàm tính khoảng cách***

*Các hàm tính khoảng cách hình học*

* Hàm Minkowski (p-norm)



* Hàm Mahattan (p=1)



* Hàm Euclid (p=2)



*Hàm khoảng cách Hamming*

Các thuộc tính đầu vào là kiểu nhị phân ({0, 1}), ví dụ x = (0, 1, 1, 0, 1)



Với 

*Hàm tính độ tương tự Cosine*

Độ tương tự cosine được tính:



Suy ra khoảng cách giữa x và z là: 

***Chuẩn hoá miền giá trị thuộc tính***

Lấy ví dụ với hàm tính khoảng cách Euclid ở trên. Giả sử mỗi ví dụ được biểu diễn bởi 3 thuộc tính là Age (tuổi), Income (thu nhập hàng tháng), Height (chiều cao đo theo mét).





Áp dụng công thức tính khoảng cách Euclid ta có khoảng cách giữa x và z là:



Thấy rằng giá trị khoảng cách bị quyết định chủ yếu bởi giá trị khoảng cách (sự khác biệt) giữa hai ví dụ đối với thuộc tính Income. Vì thuộc tính này có miền giá trị rất lớn so với các thuộc tính còn lại.

Do đó cần phải chuẩn hoá miền giá trị (đưa về cùng một khoảng giá trị). Khoảng giá trị [0, 1] thường được sử dụng. Đối với mỗi thuộc tính .

***Trọng số của các thuộc tính***

Lấy ví dụ với hàm khoảng cách Euclid, tất cả các thuộc tính có ảnh hưởng như nhau đối với giá trị khoảng cách.

Các thuộc tính khác nhau có thể (nên) có mức độ ảnh hưởng khác nhau đối với giá trị khoảng cách. Cần phải tích hợp các giá trị trọng số của các thuộc tính trong hàm tính khoảng cách.

 với wi là trọng số của thuộc tính i

Làm sao để xác định các giá trị trọng số của các thuộc tính? Cần 2 yếu tố để xác định giá trị trọng số, yếu tố thứ nhất là dựa trên các tri thức cụ thể của bài toán (ví dụ: được chỉ định bởi các chuyên gia trong lĩnh vực của bài toán đang xét), yếu tố thứ hai là bằng một quá trình tối ưu hoá các giá trị trọng số (ví dụ: sử dụng một tập học để học một bộ các giá trị trọng số tối ưu).

***Sử dụng KNN khi nào***

Những bài toán có các đặc điểm sau sẽ phù hợp khi lựa chọn giải thuật kNN: các ví dụ được biểu diễn trong không gian vectơ Rn, số lượng các thuộc tính để biểu diễn ví dụ là không nhiều, một tập học có kích thước lớn.

Ưu điểm của giải thuật kNN: giải thuật có chi phí thấp cho quá trình huấn luyện (chỉ lưu lại các ví dụ học), hoạt động tốt với các bài toán phân loại cho n lớp do đó không cần phải học n bộ phân loại cho n lớp, việc chọn k phù hợp (thường lớn hơn nhiều so với 1) có khả năng xử lý nhiễu cao bởi phân loại/dự đoán được thực hiện dựa trên k láng giềng gần nhất.

Nhược điểm của giải thuật kNN: phải lựa chọn hàm tính khoảng các thích hợp với từng bài toán, chi phí tính toán cao (thời gian, bộ nhớ) tại thời điểm phân loại/dự đoán, có thể cho kết quả kém/sai với các thuộc tính không liên quan.

# xử lý dữ liệu

## Dữ liệu và xử lý dữ liệu

### Dữ liệu thô

Sử dụng thư viện scrapy để thu thập dữ liệu từ hai trang web: mua bán nhà nhất và wikipedia.

Tập dữ liệu được thu thập từ trang web Mua bán nhà đất (muabannhadat.vn) chuyên cho đăng tin mua bán nhà đất. Dữ liệu ban đầu khi thu thập từ trang web về bao gồm khoảng 100 nghìn bản tin, chứa nhiều trường dữ liệu khác nhau, lưu vào một tệp tin có độ lớn 367 MB

Một tập dữ liệu khác về phân bố dân cư của toàn bộ các huyện, thị xã, thành phố thuộc các tỉnh, thành phố trên đất nước Việt Nam, tệp dữ liệu này được sử dụng cho quá trình xử lý dữ liệu ở phần tiếp theo. Dữ liệu được thu thập từ trang web Wikipedia gồm 701 huyện, thị xã thuộc 63 tỉnh, thành phố.

### Xử lý dữ liệu

Dữ liệu thô được thu thập từ bước trên được nhóm đánh giá là xấu, nhiều bản ghi có nhiều trường dữ liệu trống, không có giá trị hoặc các giá trị không phù hợp với trường dữ liệu đó, nhiều giá trị bản ghi sai, rác, các bản ghi trùng nhau nhiều. Các bước xử lý dữ liệu dưới đây đều sử dụng công nghệ Spark để xử lý, dữ liệu trước và sau khi xử lý đều được lưu trữ bằng HDFS.

Bước 1: Lựa chọn các trường dữ liệu sử dụng để làm đầu vào của giải thuật (bao gồm Width, Length, Bedrooms, Bathrooms, District, Province). Nhãn dự đoán là trường dữ liệu Price.

Bước 2: Chỉ lấy những bản ghi có trường dữ liệu District, Province và Price. Giữ lại các bản ghi có ít nhất 1 trong các giá trị Width, Length, Bedrooms, Bathrooms khác rỗng. Bỏ qua các hàng có Price ảo (mức ảo tự quy định), loại bỏ hàng trùng.

Bước 3: Chuẩn hoá những trường có giá trị ngoại lệ (lớn hơn hay nhỏ hơn một ngưỡng xác định).

Bước 4: Điền đầy các giá trị bị thiếu.

Bước 5: Quy đổi hai trường dữ liệu District và Province ra số thực dựa trên dữ liệu thu thập được từ phân bố dân cư.

Bước 6: Chuẩn hoá miền giá trị của các trường dữ liệu về khoảng (0, 1).

### Khám phá dữ liệu

***Mô tả dữ liệu trước và sau khi xử lý***



Figure 1: Mô tả dữ liệu trước khi xử lý

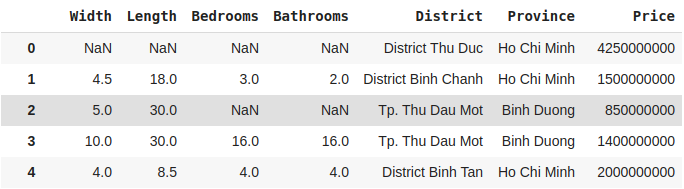


Figure 2: Một phần dữ liệu trước khi xử lý

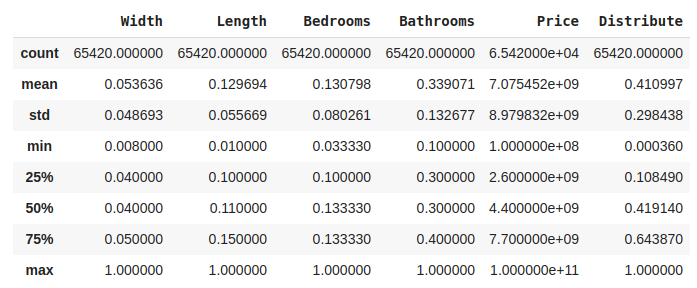


Figure 3:Mô tả dữ liệu sau khi xử lý



Figure 4: Một phần dữ liệu sau khi xử lý

*Nhận xét:* Dữ liệu sau khi xử lý xong đã loại bỏ được những giá trị không phù hợp với các trường dữ liệu. Nhưng phân bố dữ liệu vẫn còn chênh lệch, không đồng đều. Hầu hết các trường dữ liệu có giá trị gần với giá trị nhỏ nhất hơn là giá trị lớn nhất của trường đó.

***Mô tả trường dữ liệu Width, Length bằng biểu đồ***

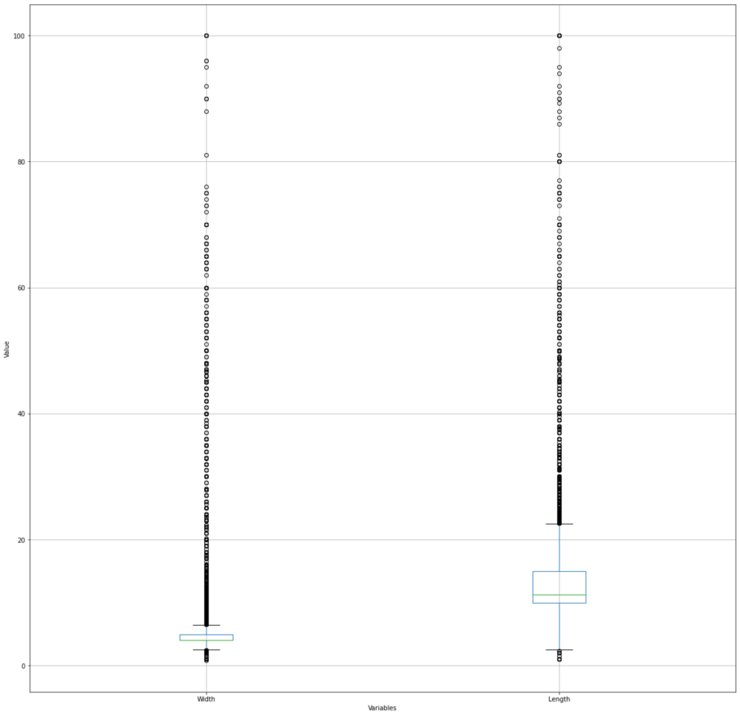
****

Figure 5: Biểu đồ mô tả Width và Length

*Nhận xét:* Hai trường dữ liệu Width và Length chứa nhiều dữ liệu ngoại biên. Vùng dữ liệu tập trung của trường Width là khoảng từ 1 đến 5. Vùng dữ liệu tập trung của trường Length là khoảng từ 1 đến 21. Từ đó cho thấy rằng những căn nhà thường có thiết kế chiều rộng hướng mặt tiền vừa đủ, khoảng 5 mét trở xuống, và chiều dài của căn nhà thường không quá dài, dưới 21 mét.

***Mô tả trường dữ liệu Bedrooms, Bathrooms bằng biểu đồ***

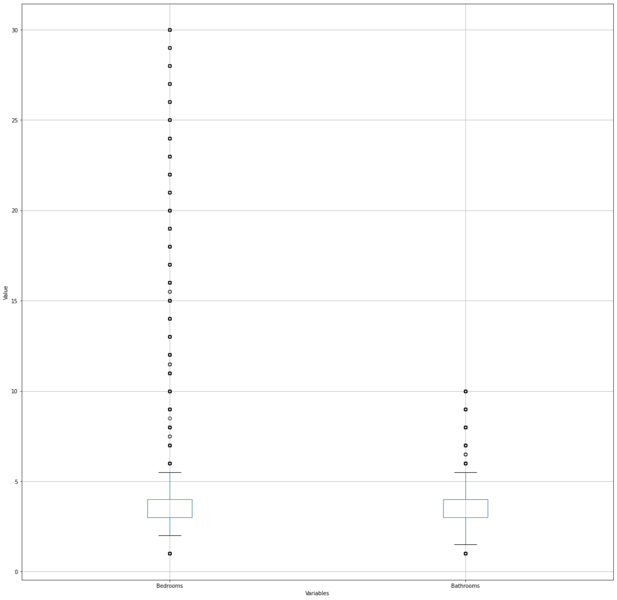
****

Figure 6: Biều đồ mô tả Bedrooms và Bathrooms

*Nhận xét:* Cả hai trường dữ liệu Bedrooms và Bathrooms vẫn có những dữ liệu ngoại biên nhưng ít hơn so với 2 trường dữ liệu trước. Vùng dữ liệu tập trung của Bedrooms là từ 2 đến 6, vùng dữ liệu tập trung của Bathrooms là từ 2 đến 6. Cho thấy các căn nhà được bán ra hầu như đều có số phòng ngủ và nhà tắm tương đối đều nhau, trung bình mỗi căn nhà bán ra có 4 phòng ngủ và 4 nhà tắm.

***Mô tả dữ liệu trường Province bằng biểu đồ***

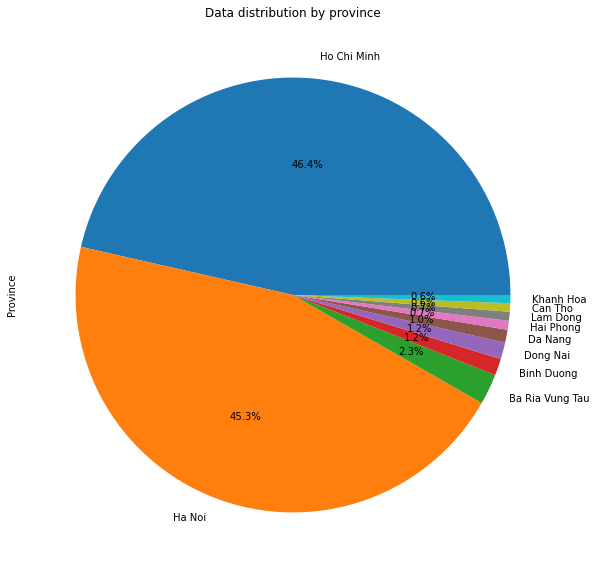
****

Figure 7: Biểu đồ quạt mô tả dữ liệu Province

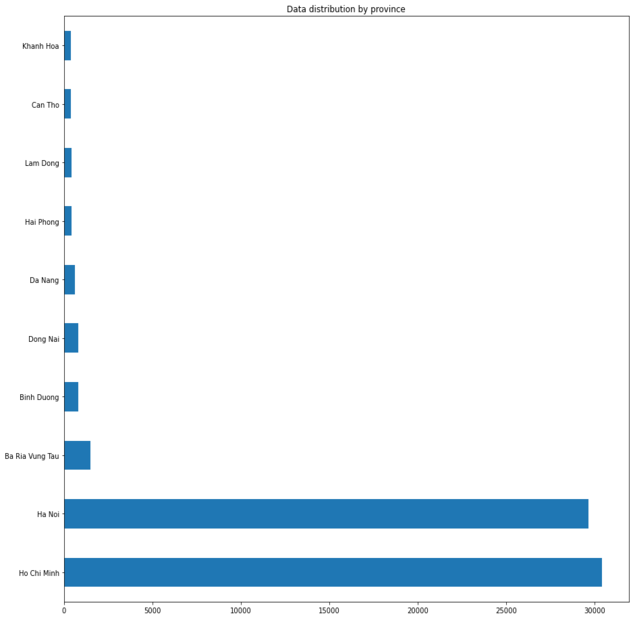


Figure 8: Biều đồ cột mô tả trường Province

*Nhận xét:* Số lượng lớn các bản tin đăng bán nhà hầu hết là ở hai thành phố là Hà Nội và Thành phố Hồ Chí Minh, sau đó là các tỉnh thành khác. Điều này cũng dễ giải thích là bởi vì Hà Nội và Thành phố Hồ Chí Minh là hai thành phố trọng điểm, trung tâm của cả nước. Vì vậy mọi người sẽ tập trung ở hai thành phố này để sống và làm việc. Các tỉnh thành phố tiếp theo có thể nhìn được ở biểu đồ trên là những tỉnh thành phố cũng nổi tiếng, có thành phố trực thuộc trung ương, vì vậy cơ hội việc làm ở đây cũng rất cao, nên vấn đề nhà đất cũng rất “nóng”.

## Phương pháp đánh giá giải thuật

Để đánh giá và so sánh giữa hai giải thuật: hồi quy tuyến tính và K láng giềng gần nhất với nhau. Nhóm sử dụng hai phương pháp đánh giá là mean square error và r2 score của thư viện sklearn.

### Mean Square Error



Với là giá trị thực tế,  là giá trị dự đoán

Giá trị MSE càng nhỏ thì giải thuật và bộ tham số chọn càng tốt và ngược lại.

### R2 Score



Với là giá trị thực tế,  là giá trị dự đoán

Giá trị R2 Score luôn có giá trị nhỏ hơn hoặc bằng 1. Giá trị này càng gần với 1 thì giải thuật và bộ tham số chọn càng tốt và ngược lại.

## Thực nghiệm

Chia tập dữ liệu đã được xử lý thành 2 tập train và test, tập train gồm 51400 bản tin, tập test gồm 12851 bản tin.

### Hồi quy tuyến tính

Thực hiện đưa dữ liệu train vào giải thuật hồi quy tuyến tính sau đó dự đoán trên tập test thu được kết quả y\_test. Kết quả được:

*Mean squared error*: 55862855033013207040.00

*R2\_score*: 0.23

### K láng giềng gần nhất

Thực hiện đưa dữ liệu train vào giải thuật với nhiều bộ tham số khác nhau. Kết quả như trong bảng:

(Với weights là một trong 2 giá trị uniform và distance, uniform thể hiện rằng ảnh hưởng của k láng giềng đối với mục đích dự đoán là như nhau, còn distance thể hiện ảnh hưởng của k láng giềng phụ thuộc vào khoảng cách, mặc định của thư viện sklearn là uniform. N neighbors là số lượng k láng giềng, p = 1 sử dụng công thức khoảng cách Mahattan, p = 2 sử dụng công thức khoảng cách Euclid. MSE là giá trị mean square error, R2 score là giá trị r2 score đã giải thích ở trên)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| weights | n\_neighbors | p | MSE | R2 Score |
| uniform | 3 | 1 | 54e+18 | 0.3198 |
| 3 | 2 | 55e+18 | 0.3016 |
| 5 | 1 | 50e+18 | 0.3650 |
| 5 | 2 | 51e+18 | 0.3494 |
| 7 | 1 | 49e+18 | 0.3823 |
| 7 | 2 | 50e+18 | 0.3708 |
| 11 | 1 | 47e+18 | 0.4016 |
| 11 | 2 | 50e+18 | 0.3699 |
| 13 | 1 | 48e+18 | 0.3857 |
| 13 | 2 | 50e+18 | 0.3710 |
| 15 | 1 | 49e+18 | 0.3848 |
| 15 | 2 | 50e+18 | 0.3692 |
| 17 | 1 | 49e+18 | 0.3843 |
| 17 | 2 | 50e+18 | 0.3698 |
| Distance | 3 | 1 | 55e+18 | 0.3011 |
| 3 | 2 | 56e+18 | 0.2857 |
| 5 | 1 | 52e+18 | 0.3398 |
| 5 | 2 | 53e+18 | 0.3271 |
| 7 | 1 | 51e+18 | 0.3590 |
| 7 | 2 | 51e+18 | 0.3508 |
| 11 | 1 | 49e+18 | 0.3888 |
| 11 | 2 | 51e+18 | 0.3593 |
| 13 | 1 | 48e+18 | 0.3750 |
| 13 | 2 | 50e+18 | 0.3645 |
| 15 | 1 | 49e+18 | 0.3746 |
| 15 | 2 | 50e+18 | 0.3654 |
| 17 | 1 | 49e+18 | 0.3697 |
| 17 | 2 | 50e+18 | 0.3642 |

### Nhận xét

Từ thực nghiệm trên, thấy được rằng với dữ liệu về nhà đất như vậy, giải thuật k láng giềng gần nhất cho kết quả tốt hơn hẳn so với giải thuật hồi quy tuyến tính. Và trong giải thuật k láng giềng gần nhất, bộ tham số (weights=uniform, p=1, n\_neighbors=11) cho kết quả tốt nhất với giá trị MSE=47e+18 và R2 Score=0.4016.

Kết quả giữa hai giải thuật như vậy là bởi vì tập dữ liệu mặc dù đã được xử lý qua nhiều giai đoạn rồi, nhưng vẫn còn chứa dữ liệu bất thường và nhiễu. Do đó giải thuật đơn giản như hồi quy tuyến tính khó có thể phù hợp được với bài toán này và với tập dữ liệu này.

# Kết luận

## Kết luận chung

Qua quá trình tìm hiểu và làm việc nhóm cùng nhau, dưới sự hướng dẫn, giảng dạy của các thầy, cô giáo, nhóm chúng em đã học được rất nhiều kiến thức. Nhóm đã nắm được cách thức hoạt động của các công nghệ như scrapy để thu thập dữ liệu từ trang web, hdfs để lưu trữ dữ liệu lớn, spark để xử lý dữ liệu lớn đó phân tán, nhóm cũng đã hiểu được quy trình của một bài toán thực tế từ việc thu thập dữ liệu, lưu trữ, xử lý cho đến việc dự đoán giá trị số thực.

## Những khó khăn gặp phải

Trong quá trình tìm hiểu và làm việc nhóm cùng nhau đó, nhóm gặp phải một số vấn đề như thời gian của mỗi thành viên trong nhóm khác nhau nên thời gian đầu chưa thống nhất được lịch họp nhóm. Sau đó mỗi thành viên đã tự sắp xếp được thời gian của cá nhân và thống nhất dành một khoảng thời gian cho nhóm. Nguồn tài liệu trên internet rộng lớn, vốn từ vựng tiếng Anh của nhóm còn chưa được tốt, mặc khác tài liệu tiếng Việt thì khan hiếm. Nhưng nhóm đã cố gắng, giúp đỡ lẫn nhau, cùng với sử dụng từ điển, để khai thác tài liệu nước ngoài. Phần khó khăn nhất trong quá trình làm bài tập lớn môn học có lẽ là phần thu thập và xử lý dữ liệu. Dữ liệu thu thập từ các trang web về được nhóm đánh giá thật sự “rác”. Ban đầu nhóm có phần nhụt chí, nhưng với sự quyết tâm và động viên lẫn nhau trong nhóm, cuối cùng nhóm cũng xử lý “đủ tốt” dữ liệu để có thể sử dụng bởi các giải thuật học máy.

## Định hướng phát triển

Kết quả thực nghiệm ở trên vẫn chưa phải là kết quả mà các thành viên trong nhóm có thể “hài lòng”, nhóm em vẫn sẽ tiếp tục nghiên cứu các phương pháp xử lý dữ liệu, chuẩn hoá dữ liệu sao cho tốt hơn, phù hợp hơn với thực tế thị trường. Nhóm em cũng sẽ thu thập nhiều nguồn dữ liệu hơn nữa và kết hợp với những phân tích từ chuyên gia trong lĩnh vực bất động sản để có thể cải thiện được tốt nhất dữ liệu trước khi áp dụng các giải thuật học máy. Nhóm cũng có dự định nghiên cứu một số giải thuật học máy khác, thậm chí có thể là học sâu để áp dụng được tốt nhất dữ liệu xử lý được. Nhóm em vẫn mong nhận được sự chỉ dẫn của các thầy, cô để nhóm em hoàn thiện hơn trong tương lai.

# Tài liệu tham khảo

Tài liệu học tập, slide môn học Nhập môn khoa học dữ liệu được các thầy cô cung cấp.

Making Web Crawlers Using Scrapy for Python - <https://www.datacamp.com/community/tutorials/making-web-crawlers-scrapy-python>

HDFS Tutorial: Introduction to HDFS & its Features - <https://www.edureka.co/blog/hdfs-tutorial>

Apache Spark – Introduction - <https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm>

An introduction to simple linear regression - <https://www.scribbr.com/statistics/simple-linear-regression/>

Sklearn R2 Score - <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html>

Sklearn Mean Square Error - <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html>

A Simple Introduction to K-Nearest Neighbors Algorithm - <https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e>