

TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT
KHOA HỆ THỐNG THÔNG TIN



BÁO CÁO CUỐI KỲ

TÊN MÔN HỌC: BIG DATA VÀ ỨNG DỤNG

MÃ HỌC PHẦN: 232MI1401

**ÁP DỤNG KỸ THUẬT BIG DATA TRÊN MÔI
TRƯỜNG SPARK ĐỂ PHÂN TÍCH CÁC YẾU TỐ
ẢNH HƯỞNG ĐẾN GIÁ BẤT ĐỘNG SẢN**




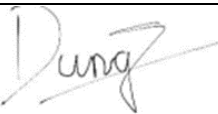

Giảng viên hướng dẫn: TS Nguyễn Thôn Dã

Danh sách thành viên nhóm:

1. K204061394, Nguyễn Thị Bảo Hà
2. K204061392, Cù Thị Mỹ Duy
3. K204061389, Nguyễn Mỹ Dung
4. K204060282, Nguyễn Thị Cẩm Giang
5. K204061390, Nguyễn Thị Mỹ Dung

Thành phố Hồ Chí Minh, 2024

Bảng tự đánh giá thành viên nhóm

STT	MSSV	Họ và tên	Điểm tự đánh giá (thang điểm 10)	Ký tên
1	K204061394	Nguyễn Thị Bảo Hà	10	
2	K204061392	Cù Thị Mỹ Duyệt	10	
3	K204061389	Nguyễn Mỹ Dung	10	
4	K204061390	Nguyễn Thị Mỹ Dung	10	
5	K204060282	Nguyễn Thị Cẩm Giang	10	

Lời cảm ơn của nhóm

Nhóm chúng em rất biết ơn những kiến thức quý báu mà nhóm đã có được từ môn học: Big Data và Ứng dụng, những kiến thức này đã góp phần giúp nhóm hoàn thành thành công dự án của mình. Nhóm xin gửi lời cảm ơn chân thành tới các giảng viên đã hướng dẫn và hỗ trợ nhiệt tình trong suốt khóa học.

Nhóm xin gửi lời cảm ơn chân thành đến thầy Nguyễn Thôn Dã, người đã giúp đỡ chúng em rất nhiều trong việc có được những kỹ năng và kiến thức cần thiết cũng như đưa ra những gợi ý và giải pháp để xây dựng và phát triển đồ án.

Dù nhóm đã cố gắng nỗ lực và thực hiện dự án hết khả năng nhưng do kiến thức và kinh nghiệm chuyên môn hạn chế của nhóm vẫn còn tồn tại sai sót trong cách trình bày, thực hiện và đánh giá vấn đề. Vì vậy, nhóm rất mong nhận được những phản hồi, đánh giá từ thầy để nâng cao hiểu biết và nâng cao chất lượng của đồ án.

Lời cam kết

Chúng tôi cam đoan kết quả nghiên cứu này là của riêng chúng tôi, chúng tôi khẳng định không sao chép kết quả nghiên cứu của những cá nhân hoặc nhóm nghiên cứu nào khác.

Ho Chi Minh City, 01/2024

Tập thể thành viên nhóm

Mục lục

Bảng tự đánh giá thành viên nhóm.....	1
Lời cảm ơn của nhóm	2
Lời cam kết	3
Mục lục	4
Danh mục hình ảnh.....	6
CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI	7
1.1 Giới thiệu tổng quan	7
CHƯƠNG 2: ĐỀ XUẤT MÔ HÌNH.....	8
CHƯƠNG 3: CƠ SỞ LÝ THUYẾT.....	10
3.1 Mã hóa	10
3.2 Thuật toán: Decision tree, Random forest, XGboost	10
3.2.1 Decision tree	10
3.2.2 Random forest	11
3.2.3 XGboost.....	12
3.2.4 Mô hình linear regression.....	14
3.3 Các chỉ số đánh giá mô hình	14
CHƯƠNG 4: THU THẬP VÀ TIỀN XỬ LÝ DỮ LIỆU	16
4.1 Thu thập và mô tả dữ liệu.....	16
4.2 Thư viện sử dụng.....	18
4.3 Tiền xử lý dữ liệu	18
CHƯƠNG 5: PHÂN TÍCH VÀ TRỰC QUAN HÓA DỮ LIỆU	22
5.1 Phân tích khám phá dữ liệu	22
5.1.1 Kiểm tra sự phân bố của dữ liệu.....	22
5.1.2 Phân tích biến Price	23
5.1.3 Price theo biến Total rooms.....	24
5.1.4 Phân tích biến Price theo thu nhập bình quân đầu người.....	25
5.1.5 Phân tích biến Price theo biến Ocean Proximity	26
5.1.6 Price theo vị trí địa lý	29
5.1.7 Phân tích biến Housing Median Age.....	31

5.2 Ma trận tương quan	32
CHƯƠNG 6: THỰC NGHIỆM VÀ ĐÁNH GIÁ MÔ HÌNH	33
6.1 Feature Engineering.....	33
6.1.1 Thông tin các đặc trưng.....	33
6.1.2 Encoding.....	33
6.2 Phân tập Train, Test và thực nghiệm các thuật toán	35
6.3 Kết quả thực nghiệm và thảo luận.....	35
CHƯƠNG 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	37
7.1 Kết quả đạt được.....	37
7.2 Hạn chế.....	37
7.3 Hướng phát triển.....	37

Danh mục hình ảnh

Hình 3. 1 Mô hình minh họa decision tree	11
Hình 3. 2 Mô hình minh họa Random forest.....	12
Hình 3. 3 Mô hình minh họa XGBoost	13
Hình 5. 1 Sự phân bố của dữ liệu	22
Hình 5. 2 Phân phối chuẩn của biến Price.....	23
Hình 5. 3 Giá trị price theo biến total room	24
Hình 5. 4 Xem xét biến thu nhập trên đầu người (Per_capita_income).....	25
Hình 5. 5 Sự phân bố của biến price theo biến per_capita_income (biến thu nhập theo đầu người).....	26
Hình 5. 6 Các giá trị trong biến Ocean Proximity.....	27
Hình 5. 7 Thu nhập bình quân của cư dân theo từng giá trị trong biến Ocean_proximity.	28
Hình 5. 8 Sự phân bố theo vị trí địa lý (2 biến longitude và latitude).....	29
Hình 5. 9 Trực quan hóa trên bản đồ.....	30
Hình 5. 10 Sự phân bố của các giá trị trong biến housing_median_age	31
Hình 5. 11 Ma trận tương quan.	32
Hình 6. 1 Kiểm tra loại dữ liệu của các cột dữ liệu.....	33
Hình 6. 2 Chuyển đổi dữ liệu của cột “ocean_proximity” từ dạng chữ sang dạng số.	33
Hình 6. 3 Chuyển dữ liệu và gán chỉ mục cho các giá trị thuộc các cột.	34
Hình 6. 4 Đưa dữ liệu về dạng vecto tổng hợp và phân tập train,test	34
Hình 6. 5 Các thuật toán để thực nghiệm mô hình.	35

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

1.1 Giới thiệu tổng quan

Trong bối cảnh một thế giới ngày càng số hóa và thu thập dữ liệu ngày một lớn mạnh, việc áp dụng kỹ thuật Big Data trên môi trường Spark trở thành một xu hướng quan trọng để tận dụng thông tin đa dạng từ nhiều nguồn khác nhau. Trong lĩnh vực bất động sản, nơi mà giá trị của mỗi ngôi nhà không chỉ phản ánh đặc điểm vật lý mà còn phụ thuộc vào nhiều yếu tố ảnh hưởng khác nhau, việc sử dụng Big Data vào môi trường Spark là chìa khóa để mở ra những hiểu biết sâu sắc về thị trường.

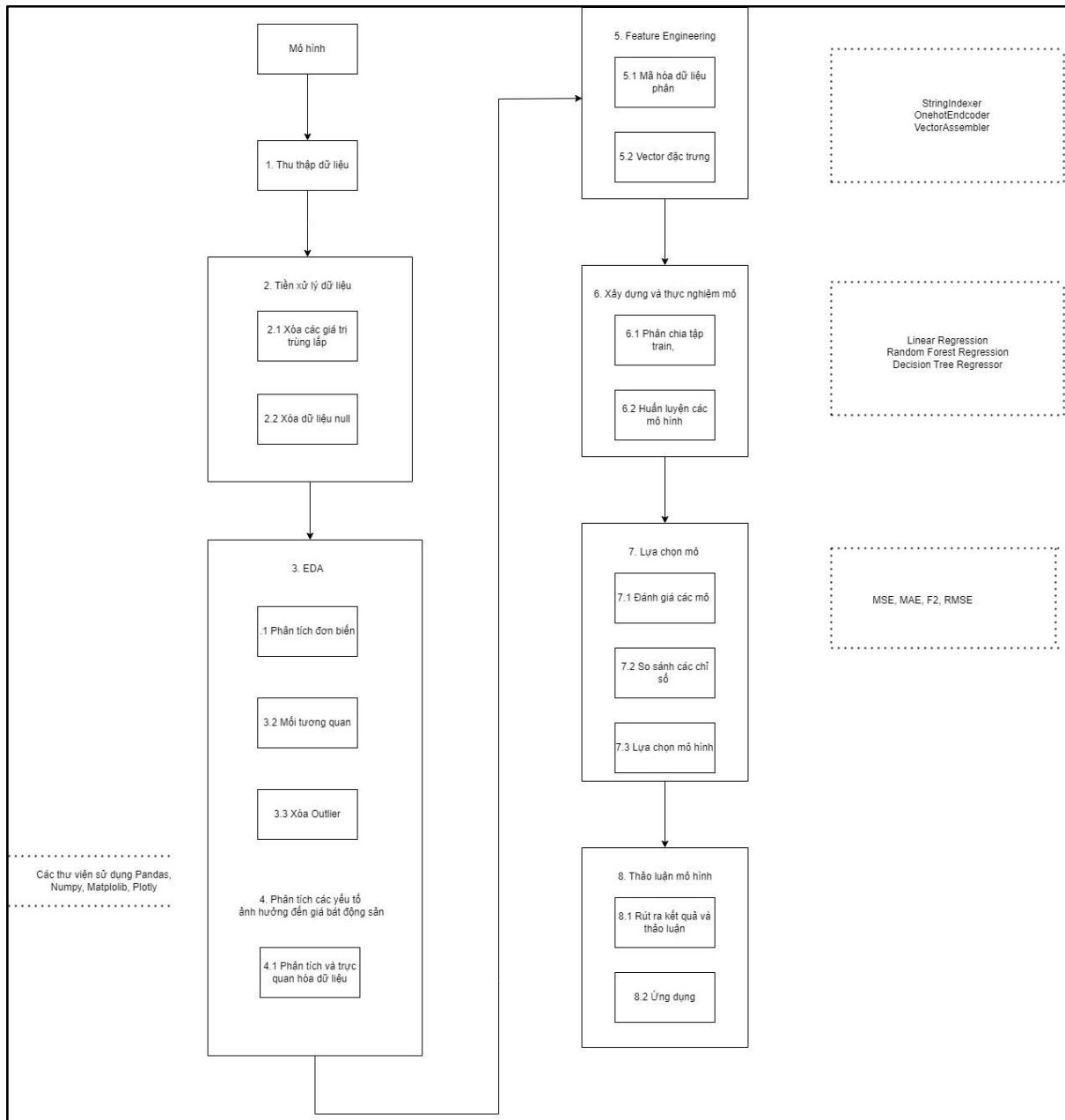
Mục tiêu chính của đề tài này là xây dựng một mô hình học máy chất lượng cao, có khả năng dự đoán giá trị ngôi nhà trung bình dựa trên các đặc điểm nhất định. Qua việc sử dụng dữ liệu lớn được thu thập từ nhiều nguồn như vị trí địa lý, cơ sở hạ tầng, môi trường xã hội và kinh tế, mô hình này sẽ cung cấp cái nhìn toàn diện về yếu tố ảnh hưởng đến giá bất động sản.

Dự đoán giá nhà thông qua mô hình học máy không chỉ mang lại giá trị lớn cho người mua bán và đầu tư, mà còn đóng góp vào việc hiểu rõ hơn về xu hướng thị trường nhà ở. Bằng cách này, nó trở thành một công cụ quan trọng hỗ trợ quá trình ra quyết định liên quan đến đầu tư bất động sản, quy hoạch đô thị và phát triển chính sách.

Với việc tích hợp kỹ thuật Big Data vào môi trường Spark vào quá trình phân tích, chúng ta không chỉ nâng cao khả năng xử lý và phân tích dữ liệu mà còn tạo ra một cơ sở hạ tầng mạnh mẽ để phát triển và triển khai các ứng dụng dự đoán giá bất động sản trong thực tế. Điều này không chỉ mở ra những triển vọng mới trong nghiên cứu mà còn tạo ra những cơ hội quan trọng cho sự đổi mới trong lĩnh vực bất động sản và quản lý đô thị.

CHƯƠNG 2: ĐỀ XUẤT MÔ HÌNH

Từ những nghiên cứu trên, nhóm đã lựa chọn những thuật toán, nghiên cứu thị trường, lĩnh vực phù hợp và những đặc trưng trong để tìm kiếm bộ dữ liệu phù hợp. Từ đó, nhóm đề xuất mô hình dự đoán giá nhà thông qua mô hình học máy bằng cách áp dụng kỹ thuật Big Data trên môi trường Spark



Với mô hình nghiên cứu đề xuất ở trên, nhóm bắt đầu tiến hành thu thập dữ liệu là thông tin về giá nhà đất ở các quận của California, sau đó tiến hành làm sạch dữ liệu với các bước tiền xử lý cần thiết để xử lý các giá trị bị thiếu, các giá trị ngoại lệ và

bất kỳ sự không nhất quán dữ liệu nào khác. Bước tiếp theo là phân tích phân tích kỹ lưỡng tập dữ liệu để hiểu rõ hơn về mối quan hệ giữa các biến khác nhau và tác động của chúng đối với giá trị ngôi nhà trung bình.

Sau quá trình phân tích, nhóm thực hiện Feature Engineering để xác định các tính năng có liên quan và có khả năng tạo ra các tính năng mới có thể nâng cao khả năng dự đoán của mô hình. Các mô hình được lựa chọn để thực nghiệm và đánh giá là: Linear regression, Random forest regression, Decision tree regression. Các chỉ số: giá trị sai số tuyệt đối trung bình (MAE), sai số bình phương trung bình (MSE), sai số bình phương trung bình gốc (RMSE) và bình phương R được sử dụng để so sánh kết quả các mô hình. Cuối cùng, nhóm sẽ lựa chọn mô hình có độ tin cậy tốt để dự đoán giá nhà, đưa ra các thảo luận và ứng dụng trong thực tế.

CHƯƠNG 3: CƠ SỞ LÝ THUYẾT

3.1 Mã hóa

Mã hóa là quá trình chuyển đổi dữ liệu thô thành dạng có thể hiểu được và sử dụng được bởi các thuật toán học máy. Dữ liệu thô thường ở dạng không định cấu trúc hoặc khó hiểu, khiến cho việc học máy trở nên khó khăn. Mã hóa giúp biến đổi dữ liệu thành dạng số hoặc dạng vector, giúp cho các thuật toán học máy có thể dễ dàng xử lý và phân tích:

- Mã hóa One-hot: Dữ liệu danh mục (categorical data) như màu sắc, quốc gia, v.v. thường được mã hóa bằng cách sử dụng mã hóa one-hot. Ví dụ, dữ liệu về màu sắc ("đỏ", "xanh lam", "vàng") có thể được mã hóa thành ba vector nhị phân $(0, 1, 0)$, $(0, 0, 1)$, $(1, 0, 0)$.
- Mã hóa Label Encoding: Dữ liệu danh mục có thể được mã hóa thành các số nguyên. Ví dụ, dữ liệu về quốc gia ("Việt Nam", "Mỹ", "Nhật Bản") có thể được mã hóa thành các số 1, 2, 3.

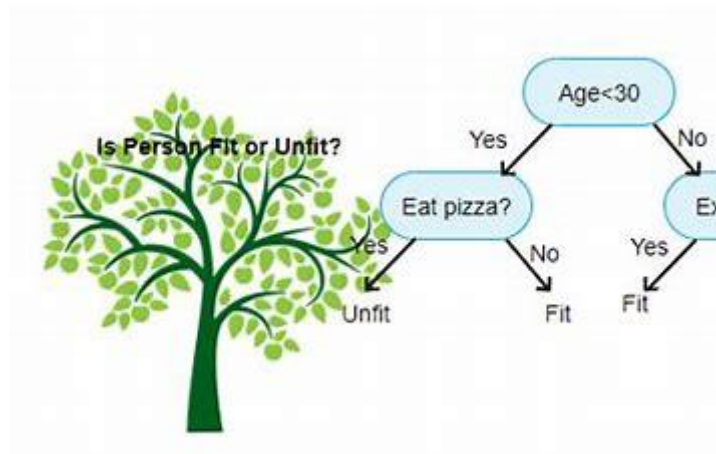
Mã hóa giúp cho các thuật toán học máy dễ dàng xử lý và phân tích dữ liệu, Cải thiện hiệu suất của các mô hình học máy, Giúp cho việc giải thích mô hình học máy trở nên dễ dàng hơn. Mã hóa được dùng trong phân loại văn bản, dự đoán giá trị, xác định đối tượng. Mã hóa là một bước quan trọng trong quá trình học máy. Việc lựa chọn phương pháp mã hóa phù hợp sẽ phụ thuộc vào loại dữ liệu và mục tiêu của mô hình học máy.

3.2 Thuật toán: Decision tree, Random forest, XGboost

3.2.1 Decision tree

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary), Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.

Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.



Hình 3. 1 Mô hình minh họa decision tree

Ví dụ:

Cây quyết định này dự đoán liệu một người đàn ông có đi đá bóng hay không dựa trên thời tiết, độ ẩm và gió.

Thuộc tính:

- + Thời tiết: Có thể là "Nắng", "Mưa" hoặc "Mây"
- + Độ ẩm: Có thể là "Cao" hoặc "Thấp"
- + Gió: Có thể là "Mạnh" hoặc "Yếu"

Mục tiêu: Dự đoán liệu người đàn ông có đi đá bóng hay không.

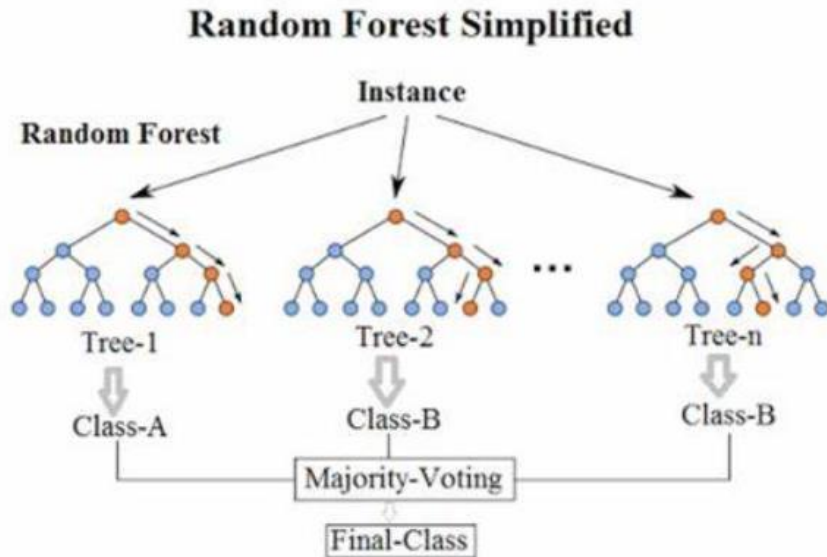
→ Cây quyết định này là một công cụ đơn giản có thể được sử dụng để dự đoán liệu một người đàn ông có đi đá bóng hay không dựa trên thời tiết, độ ẩm và gió.

3.2.2 Random forest

Là một thuật toán học máy được sử dụng để phân loại và hồi quy. Nó hoạt động bằng cách tạo ra một rừng các cây quyết định và sau đó sử dụng kết quả của các cây đó để đưa ra dự đoán.

Cách thức hoạt động của rừng ngẫu nhiên:

- + Một tập dữ liệu được sử dụng để tạo ra một rừng các cây quyết định.
- + Mỗi cây quyết định được tạo ra bằng cách sử dụng một tập hợp con ngẫu nhiên của dữ liệu và một tập hợp con ngẫu nhiên của các tính năng.
- + Mỗi cây quyết định đưa ra dự đoán cho mỗi điểm dữ liệu.
- + Các dự đoán từ các cây quyết định được kết hợp để đưa ra dự đoán cuối cùng.



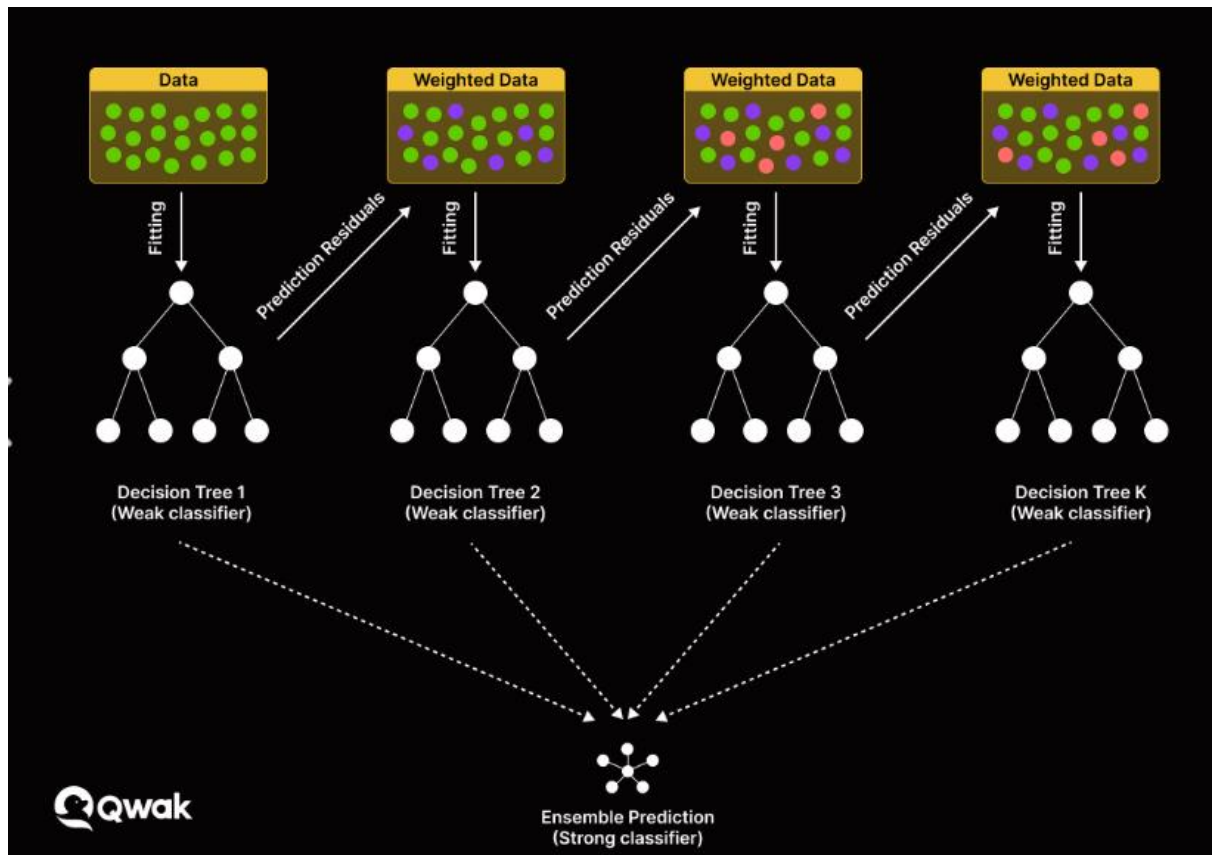
Hình 3. 2 Mô hình minh họa Random forest

Ưu điểm của rừng ngẫu nhiên: Chính xác, có thể xử lý dữ liệu có nhiễu, có thể xử lý dữ liệu có nhiều tính năng, Có thể được sử dụng cho cả phân loại và hồi quy. Nhược điểm của rừng ngẫu nhiên: có thể tốn thời gian để đào tạo, có thể khó diễn giải.

Kết luận: Rừng ngẫu nhiên là một thuật toán học máy mạnh mẽ có thể được sử dụng cho nhiều mục đích khác nhau. Nó chính xác, có thể xử lý dữ liệu có nhiễu và có thể xử lý dữ liệu có nhiều tính năng. Tuy nhiên, nó có thể tốn thời gian để đào tạo và có thể khó diễn giải.

3.2.3 XGboost

Hoạt động dựa trên ý tưởng tăng cường (boosting) nhiều mô hình cây quyết định (decision tree) yếu để tạo ra một mô hình mạnh mẽ hơn. Các mô hình cây quyết định được thêm vào mô hình tổng thể một cách tuần tự, với mỗi mô hình mới được xây dựng để sửa lỗi của các mô hình trước đó.



Hình 3. 3 Mô hình minh họa XGBoost

Ưu điểm:

- + Chính xác cao: XGBoost thường đạt được độ chính xác cao hơn các thuật toán học máy khác.
- + Hiệu quả: XGBoost có tốc độ đào tạo nhanh và sử dụng bộ nhớ hiệu quả.
- + Khả năng mở rộng: XGBoost có thể xử lý các tập dữ liệu lớn.
- + Dễ sử dụng: XGBoost có nhiều thư viện và công cụ hỗ trợ cho nhiều ngôn ngữ lập trình khác nhau.

Nhược điểm:

- + Có thể bị quá khớp: XGBoost có thể bị quá khớp với dữ liệu đào tạo nếu không được điều chỉnh cẩn thận.
- + Khó diễn giải: XGBoost có thể khó diễn giải hơn các mô hình học máy đơn giản khác.

Ứng dụng:

XGBoost được sử dụng trong nhiều lĩnh vực khác nhau, bao gồm:

- + Phân loại ảnh: XGBoost có thể được sử dụng để phân loại các đối tượng trong ảnh.
- + Xử lý ngôn ngữ tự nhiên: XGBoost có thể được sử dụng để phân loại văn bản và trích xuất thông tin.
- + Khuyến nghị: XGBoost có thể được sử dụng để đề xuất các sản phẩm hoặc dịch vụ cho người dùng.
- + Phát hiện gian lận: XGBoost có thể được sử dụng để phát hiện các giao dịch gian lận.

XGBoost là một thuật toán học máy mạnh mẽ và linh hoạt có thể được sử dụng cho nhiều mục đích khác nhau. Nó là một lựa chọn tốt cho các bài toán phân loại và hồi quy đòi hỏi độ chính xác cao và hiệu quả.

3.2.4 Mô hình linear regression

Mô hình linear regression, hay hồi quy tuyến tính, là một phương pháp thống kê được sử dụng để mô hình hóa mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Mô hình này sử dụng một hàm tuyến tính để dự đoán giá trị của biến phụ thuộc dựa trên giá trị của các biến độc lập.

Mô hình linear regression có một số giả định, chẳng hạn như:

Mối quan hệ giữa biến phụ thuộc và các biến độc lập là tuyến tính. Các biến độc lập không có mối quan hệ tuyến tính nào với nhau. Sai số của mô hình là ngẫu nhiên và có phân phối chuẩn.

Nếu các giả định này không được thỏa mãn, thì mô hình linear regression có thể cung cấp kết quả không chính xác.

Mô hình linear regression được sử dụng rộng rãi trong nhiều lĩnh vực, chẳng hạn như: Kinh doanh: dự đoán doanh số, lợi nhuận, giá cả, v.v.

Khoa học: nghiên cứu y học, nghiên cứu thị trường, v.v. Kỹ thuật: kiểm soát chất lượng, dự báo thời tiết, v.v.

Mô hình linear regression là một công cụ mạnh mẽ có thể được sử dụng để giải quyết nhiều vấn đề thực tế. Tuy nhiên, điều quan trọng là phải hiểu các giả định của mô hình và các hạn chế của mô hình trước khi sử dụng nó.

3.3 Các chỉ số đánh giá mô hình

Độ lệch tuyệt đối trung bình (MAE)

Là thước đo độ sai lệch giữa giá trị dự đoán và giá trị thực tế trong thống kê và học máy. Nó là tổng độ lệch tuyệt đối giữa các giá trị dự đoán và giá trị thực tế chia cho số lượng quan sát.

Công thức:

$$MAE = (1/n) * \sum |y_i - \hat{y}_i|$$

Trong đó: n là số lượng quan sát, y_i là giá trị thực tế của quan sát thứ i, \hat{y}_i là giá trị dự đoán của quan sát thứ i.

MAE là một thước đo lỗi có thể diễn giải được và không nhạy cảm với các giá trị ngoại lệ. Nó là một lựa chọn tốt cho các mô hình dự đoán giá trị liên tục.

Sai số bình phương trung bình (MSE)

Là thước đo độ sai lệch giữa giá trị dự đoán và giá trị thực tế trong thống kê và học máy. Nó là tổng bình phương sai số giữa các giá trị dự đoán và giá trị thực tế chia cho số lượng quan sát.

Công thức cho MSE là:

$$MSE = (1/n) * \sum (y_i - \hat{y}_i)^2$$

Trong đó: n là số lượng quan sát, y_i là giá trị thực tế của quan sát thứ i, \hat{y}_i là giá trị dự đoán của quan sát thứ i

MSE là một thước đo lỗi hữu ích có thể được sử dụng để đánh giá hiệu suất của mô hình hồi quy. Nó là một thước đo có thể diễn giải được và nhạy cảm với các giá trị ngoại lệ.

Lỗi bình phương trung bình căn bậc hai (RMSE):

Là thước đo độ sai lệch giữa giá trị dự đoán và giá trị thực tế trong thống kê và học máy. Nó là căn bậc hai của sai số bình phương trung bình (MSE).

Công thức cho RMSE là:

$$RMSE = \sqrt{MSE}$$

RMSE là một thước đo lỗi hữu ích có thể được sử dụng để đánh giá hiệu suất của mô hình hồi quy. Nó là một thước đo có thể diễn giải được và có cùng đơn vị đo với dữ liệu gốc.

CHƯƠNG 4: THU THẬP VÀ TIỀN XỬ LÝ DỮ LIỆU

4.1 Thu thập và mô tả dữ liệu

Bộ dữ liệu được tìm kiếm trên Kaggle bao gồm các thông tin liên quan đến dự đoán giá nhà đất ở các quận của California. Nội dung dữ liệu bao gồm thông tin các yếu tố ảnh hưởng đến giá nhà đất.

Link bộ dữ liệu: <https://www.kaggle.com/datasets/camnugent/california-housing-prices>

Tập dữ liệu gồm có 20640 dòng dữ liệu với 10 trường dữ liệu:.

- Longitude: Tọa độ địa lý thể hiện vị trí Đông Tây của quận.
- Latitude: Tọa độ địa lý thể hiện vị trí Nam Bắc của quận.
- Housing Median Age: Tuổi trung bình của nhà ở trên địa bàn quận.
- Total Rooms: Tổng số phòng trong tất cả các ngôi nhà trong quận.
- Total Bedrooms: Tổng số phòng ngủ của tất cả các ngôi nhà trong quận.
- Population: Tổng dân số của quận.
- Households: Tổng số hộ trong quận.
- Median Income: Thu nhập trung bình của các hộ gia đình trong quận
- Median House Value: Giá trị trung bình của các ngôi nhà trong quận, được coi là biến mục tiêu để dự đoán.
- Ocean Proximity: Vị trí gần biển của quận (biển phân loại).

Các hàng dữ liệu:

longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

Các cột dữ liệu trên có kiểu dữ liệu như sau:

	Column Name	Data type
0	longitude	double
1	latitude	double
2	housing_median_age	double
3	total_rooms	double
4	total_bedrooms	double
5	population	double
6	households	double
7	median_income	double
8	median_house_value	double
9	ocean_proximity	string

⇒ Từ những thông tin cơ bản về nhà ở bao gồm giá nhà và các thông tin ảnh hưởng đến giá nhà đất ở các quận của California, giá nhà đất sẽ được dự báo và được kiểm tra bằng các chỉ số chính xác.

4.2 Thư viện sử dụng

```
[ ] # import libraries
import pyspark
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from pyspark.sql import SparkSession
from pyspark.sql.types import * # to create a dataframe of a specific type
from pyspark.sql.functions import * # importig SQL functions
from pyspark.sql.window import Window
from pyspark.sql.functions import col, lit

import os

from pyspark.sql import SparkSession, SQLContext

import pyspark.sql.functions as F
from pyspark.sql.functions import udf, col

from pyspark.ml.regression import LinearRegression
from pyspark.mllib.evaluation import RegressionMetrics

from pyspark.ml.tuning import ParamGridBuilder, CrossValidator, CrossValidatorModel
from pyspark.ml.feature import VectorAssembler, StandardScaler
from pyspark.ml.evaluation import RegressionEvaluator
```

4.3 Tiền xử lý dữ liệu

Đọc tệp dữ liệu

```
# Read the datasets
df_housing = spark.read.csv("/content/drive/MyDrive/NĂM 4/Big Data/Housingprice/housing.csv",
inferSchema=True, header=True)
```

- Xem kiểu dữ liệu của từng cột dữ liệu:

```
# print the schema of the dataset
df_housing.printSchema()

root
 |-- longitude: double (nullable = true)
 |-- latitude: double (nullable = true)
 |-- housing_median_age: double (nullable = true)
 |-- total_rooms: double (nullable = true)
 |-- total_bedrooms: double (nullable = true)
 |-- population: double (nullable = true)
 |-- households: double (nullable = true)
 |-- median_income: double (nullable = true)
 |-- median_house_value: double (nullable = true)
 |-- ocean_proximity: string (nullable = true)
```

- Đếm hàng và cột trong tập dữ liệu:

```
# count the number of rows and columns in the dataset
print((df_housing.count(), len(df_housing.columns)))

(20640, 10)
```

- Kiểm tra giá trị null trong tập dữ liệu hạng vé Phổ thông:

```
# Check null values
df_housing.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in df_housing.columns]).show()

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|households|median_income|median_house_value|ocean_proximity|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      0|      0|              0|      0|          207|      0|      0|          0|          0|              0|
```

- Loại bỏ các giá trị null:

```
# Drop the null values
df_housing = df_housing.dropna()
```

- Kiểm tra lại tập dữ liệu sau khi loại bỏ các giá trị null:

```
# Check null values again
df_housing.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in df_housing.columns]).show()

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|population|households|median_income|median_house_value|ocean_proximity|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|      0|      0|              0|      0|          0|      0|      0|          0|          0|              0|
```

- Kiểm tra các giá trị trùng lặp trong tập dữ liệu:

#Check duplicates

```
df_housing.count() - df_housing.dropDuplicates().count()
```

0

- Kiểm tra các giá trị bất thường trong tập dữ liệu:

```
# Describe the dataset  
df_housing.describe().show()
```

summary	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
count	20433	20433	20433	20433	20433	20433	20433	20433	20433	20433
mean	-119.57068859198068	35.63322125972706	28.633093525179856	2636.5042333480155	537.8705525375618	1424.9469485635982	499.43346547251997	3.8711616013312273	206864.41315519012	
stddev	2.003577890751096	2.1363476663779872	12.591805202182835	2185.269566977601	421.38507007403115	1133.2084897449597	382.2992258828481	1.899291249306247	115435.66709858322	
min	-124.35	32.54	1.0	2.0	1.0	3.0	1.0	0.4999	14999.0	
max	-114.31	41.95	52.0	39320.0	6445.0	35682.0	6082.0	15.0001	500001.0	

Chúng ta có thể thấy rằng đối với Total_rooms chỉ có 20433 bản ghi, nghĩa là thiếu 207 bản ghi. rõ ràng là chúng ta có thể xử lý nó trong các giai đoạn sau.

- Xem kiểu dữ liệu của từng cột dữ liệu:

#column overview

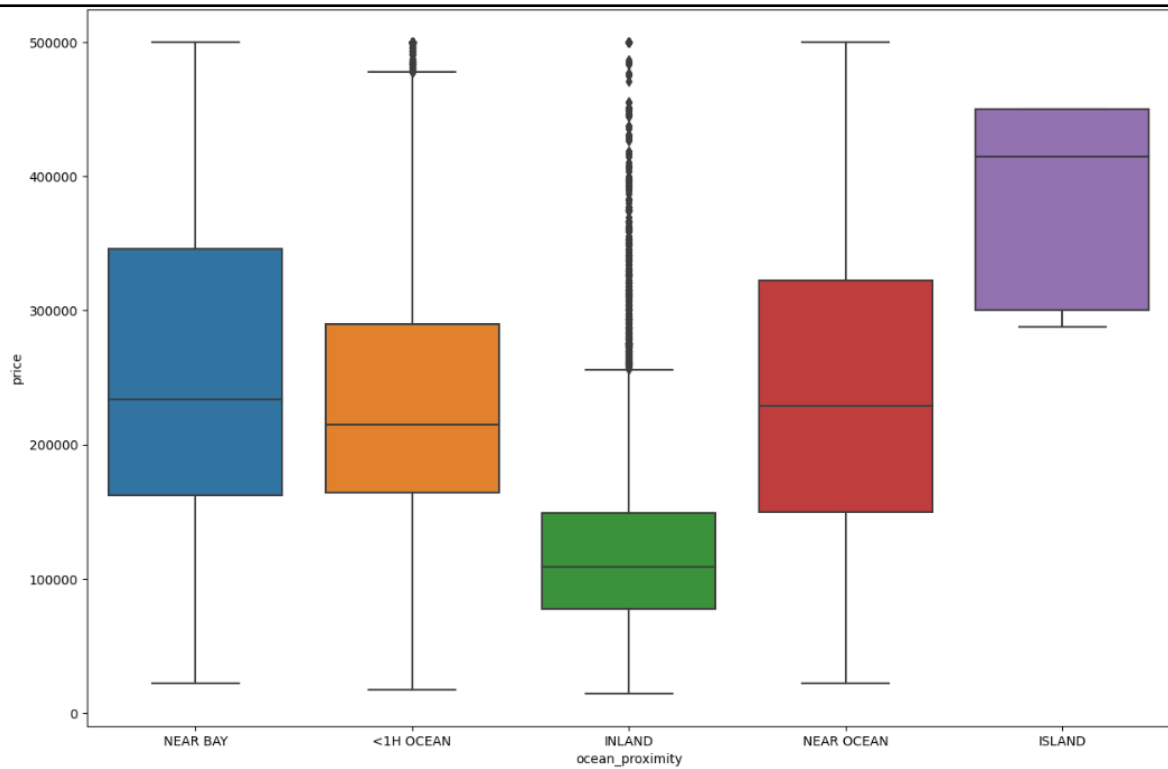
```
pd.DataFrame(df_housing.dtypes, columns = ['Column Name', 'Data type'])
```

	Column Name	Data type
0	longitude	double
1	latitude	double
2	housing_median_age	double
3	total_rooms	double
4	total_bedrooms	double
5	population	double
6	households	double
7	median_income	double
8	median_house_value	double
9	ocean_proximity	string

Khi xem xét các kiểu dữ liệu, chúng ta có thể quan sát thấy rằng tất cả các đối tượng đều có kiểu dữ liệu double nhưng chỉ có một Ocean_proximity thuộc loại string

- Kiểm tra các giá trị ngoại lai

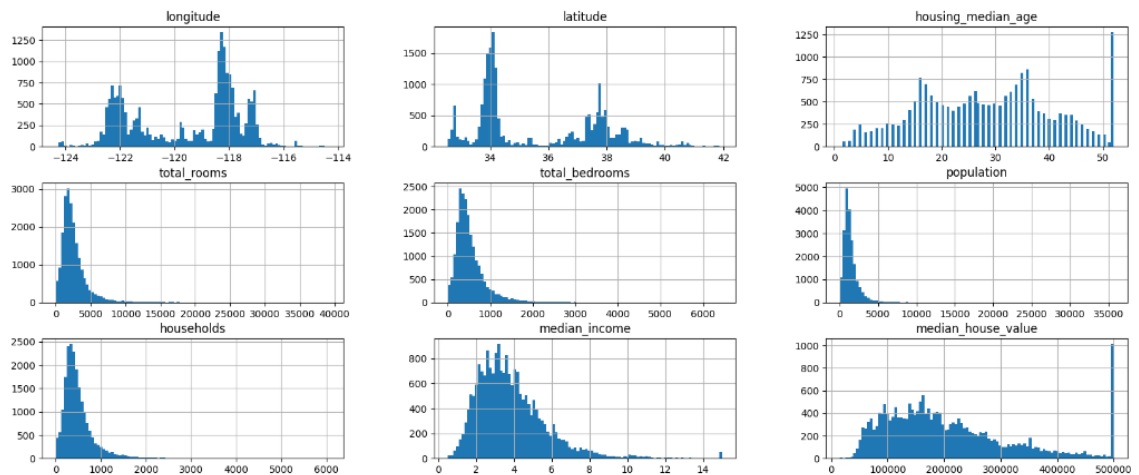
```
#visualize outliers
plt.figure(figsize=(15,10))
sns.boxplot(x='ocean_proximity', y='price', data=df_housing.toPandas())
```



CHƯƠNG 5: PHÂN TÍCH VÀ TRỰC QUAN HÓA DỮ LIỆU

5.1 Phân tích khám phá dữ liệu

5.1.1 Kiểm tra sự phân bố của dữ liệu



Hình 5. 1 Sự phân bố của dữ liệu

Bằng cách nhìn vào sự phân bố của từng dữ liệu, chúng ta có thể quan sát được một số điều:

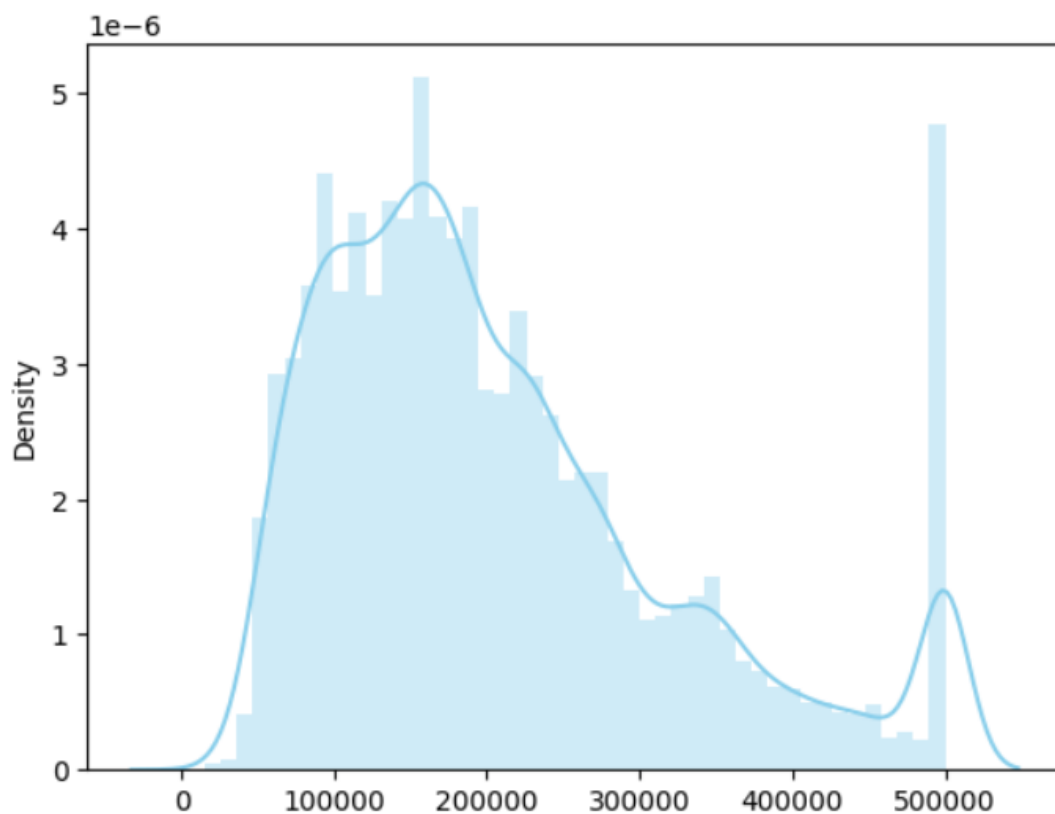
- median_income : Có vẻ như nó được giới hạn ở mức 15, tuy nhiên điều này cần xem xét khi phân tích đơn biến vì thu nhập bị giới hạn là chưa đúng..
- housing_median_age và middle_house_value : Những giá trị này cũng bị giới hạn . Điều quan trọng là phải tìm hiểu từng thuộc tính.
- Nếu chúng ta đào tạo một mô hình trực tiếp ở trên thì mô hình của chúng ta có thể hiểu rằng thu nhập sẽ không bao giờ vượt quá 15 và điều này là không đúng.
- Điều tương tự cũng có thể xảy ra với 2 biến housing_median_age và median_house_value. Bin cuối cùng cao hơn các bin gần đó, chứng tỏ chúng có khả năng các giá trị bị cắt ngọn (clip) tại các giá trị đó. Tức là các giá trị lớn hơn được chuyển thành giá trị tại điểm bị cắt.
- Các đồ thị của 2 biến về địa lý là longitude(kinh độ) và latitude (vĩ độ) tập trung nhiều điểm dữ liệu. Đây chính là kinh độ và vĩ độ địa lý của thành phố lớn tập trung nhiều đô thị vệ tinh. Đô thị vệ tinh là sự hình thành các cụm đô thị từ các cụm dân cư, chi ảnh hưởng bởi thành phố trung tâm, xây dựng các khu

nhà mới với nhiều tiện ích nhằm thu hút dân cư, phát triển dịch vụ phục vụ thành phố lớn tại các thị trấn, thị xã hiện có quanh thành phố. Theo địa lý thì bang California có khu đô thị vệ tinh là San Francisco và Los Angeles.

- Các cột `total_rooms`, `total_bedrooms`, `population`, `households` có hầu hết giá trị tập trung ở các bin đầu tiên, phần các bin sau rất dài nhưng có ít giá trị. Phân phối dữ liệu mà có dữ liệu tập trung về một phía được gọi là “đuôi dài” (long tail) hay lệch (skewed). Có thể thấy các phân phối trong trường hợp này đều là ở dạng lệch phải.

Chúng ta sẽ phải giải quyết những vấn đề này ở phần Feature Engineering.

5.1.2 Phân tích biến Price



Hình 5. 2 Phân phối chuẩn của biến Price.

Kiểm tra phân phối chuẩn của biến Price có thể thấy khoảng giá từ 100000\$ - 200000\$ là chiếm số lượng nhiều nhất.

Các tính toán liên quan đến biến Price:

Tính số căn nhà có giá trị lớn hơn 500000\$ ($\text{price} > 500000$) là 958 căn trong tổng số 20433 căn nhà trong bộ dữ liệu:

Như vậy phân khúc có giá trị cao nhất lên đến 500.000 USD chiếm tỷ lệ là 4.68%

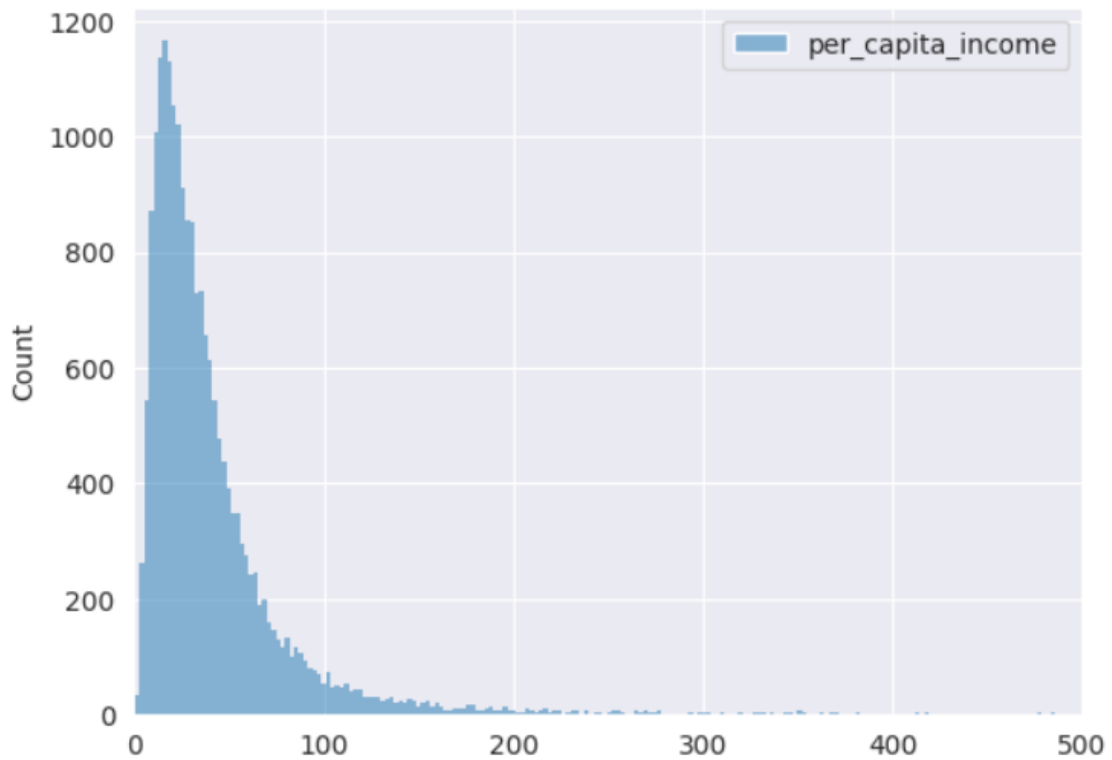
5.1.3 Price theo biến Total rooms



Hình 5. 3 Giá trị price theo biến total room

Số phòng trong khoảng từ 5.000 - 10.000 là chiếm số lượng nhiều nhất với các phân khúc giá trải dài trên giá trị của cột price. Có thể thấy tập trung nhiều nhất trong khoảng 100.000 - 300.000

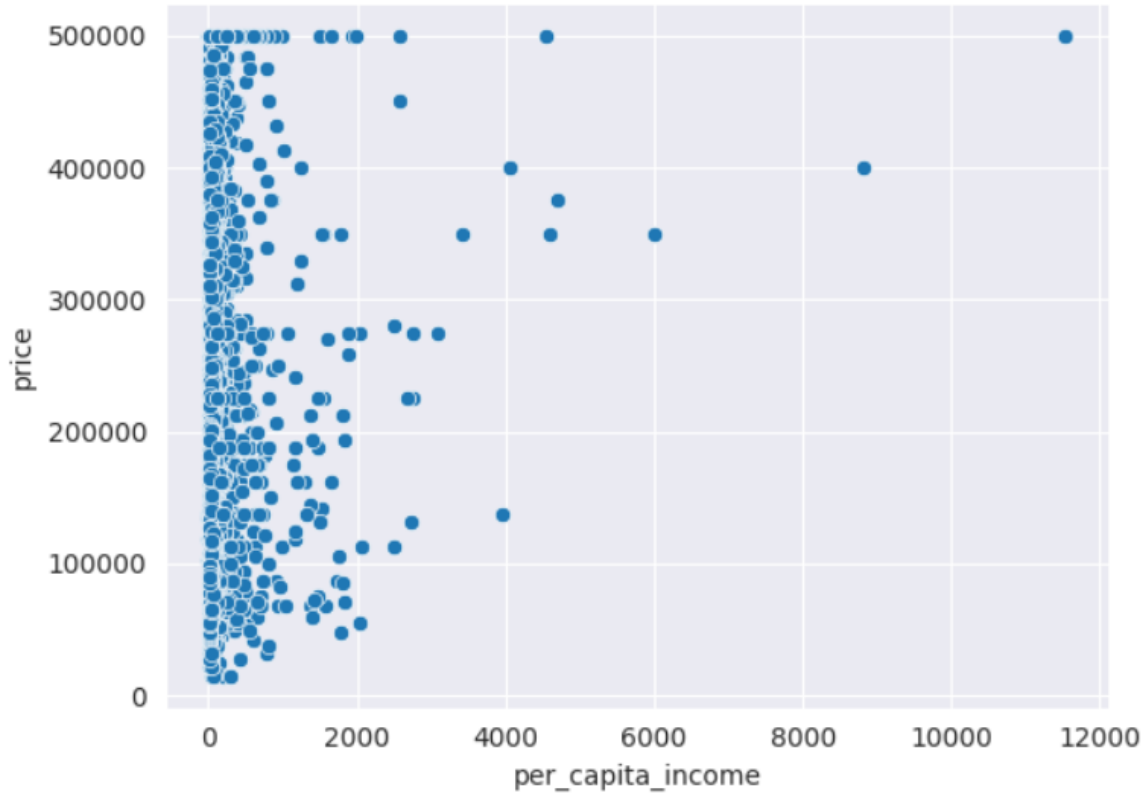
5.1.4 Phân tích biến Price theo thu nhập bình quân đầu người



Hình 5. 4 Xem xét biến thu nhập trên đầu người (*Per_capita_income*)

Nhóm thực hiện tạo thêm biến mới *Per_capita_income* bằng cách lấy Thu nhập bình quân (*median_income*) chia cho số dân trong quận (*population*).

Từ đây, chúng ta có thể xem xét được sự ảnh hưởng của thu nhập đến giá nhà đất như thế nào. Từ hình ảnh trên, cho thấy mức thu nhập của người dân rơi vào khoảng 0 - 100 là chủ yếu.



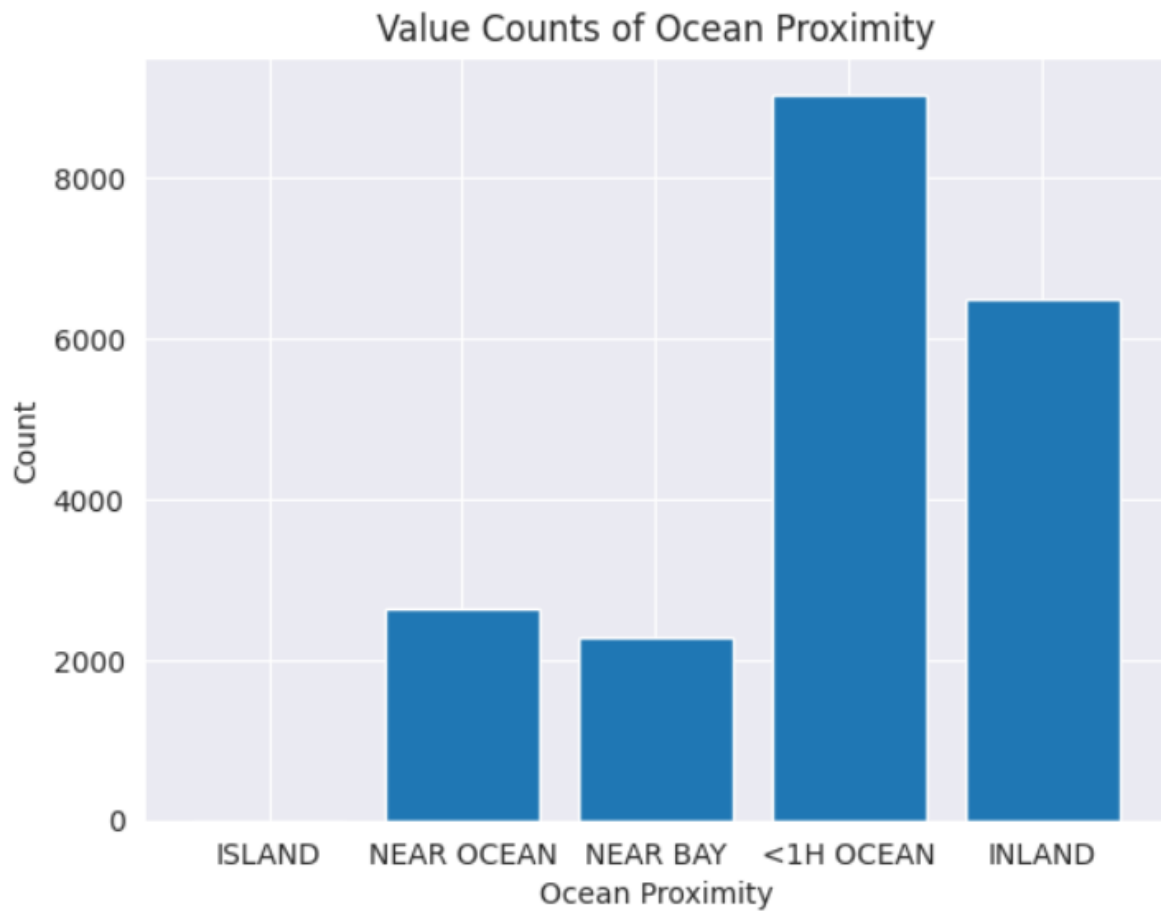
Hình 5. 5 Sự phân bố của biến price theo biến per_capita_income (biến thu nhập theo đầu người).

Phần lớn người dân có thu nhập dưới 100 USD, chiếm tỷ lệ 92.5% và giá trị bất động sản cũng phân bố tương ứng với thu nhập.

5.1.5 Phân tích biến Price theo biến Ocean Proximity

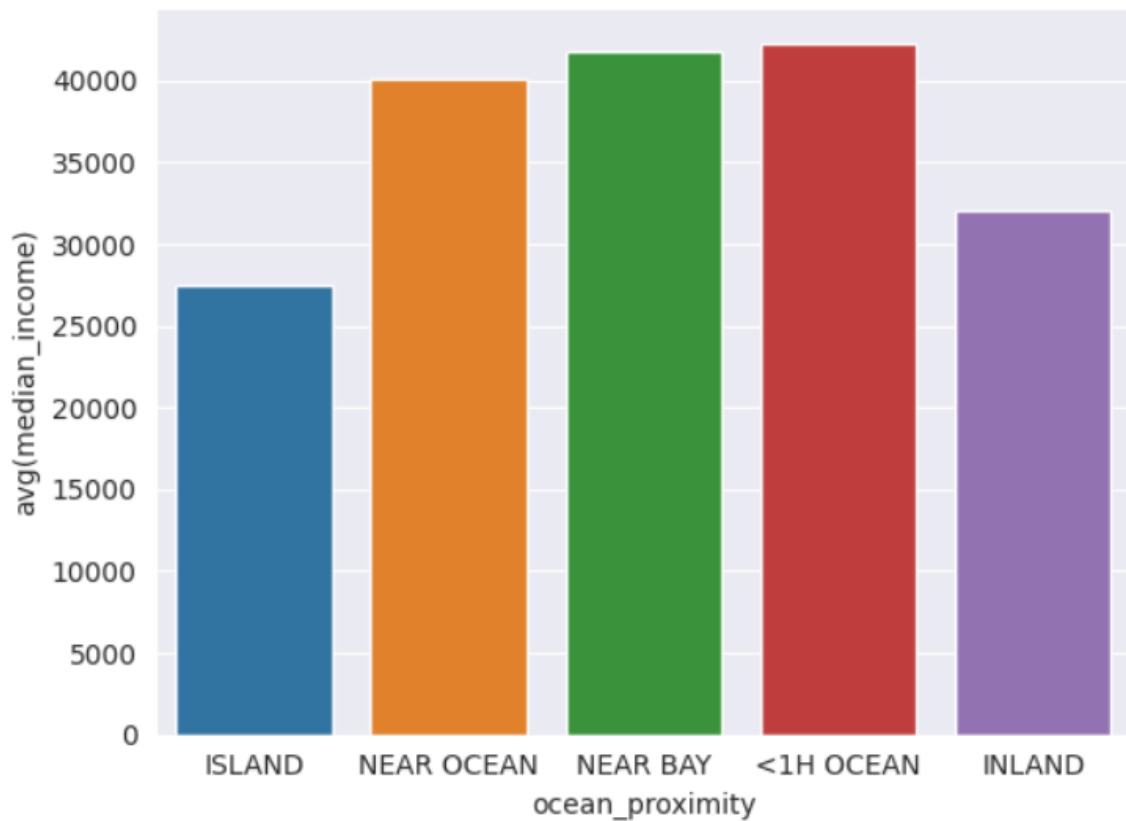
Xem xét các giá trị trong biến Ocean_Proximity: Có 5 giá trị và số lượng tương ứng như sau:

- ISLAND: có 5 giá trị
- NEAR OCEAN: có 2628 giá trị
- NEAR BAY: có 2270 giá trị
- <1H OCEAN: 9034 giá trị
- INLAND: 6496 giá trị.



Hình 5. 6 Các giá trị trong biến Ocean Proximity.

Từ đây có thể thấy dân cư sẽ tập trung nhiều nhất ở các khu vực: <1H OCEAN (khoảng cách tới biển là nhỏ hơn 1h) và INLAND (khu vực ở ngoài đảo)

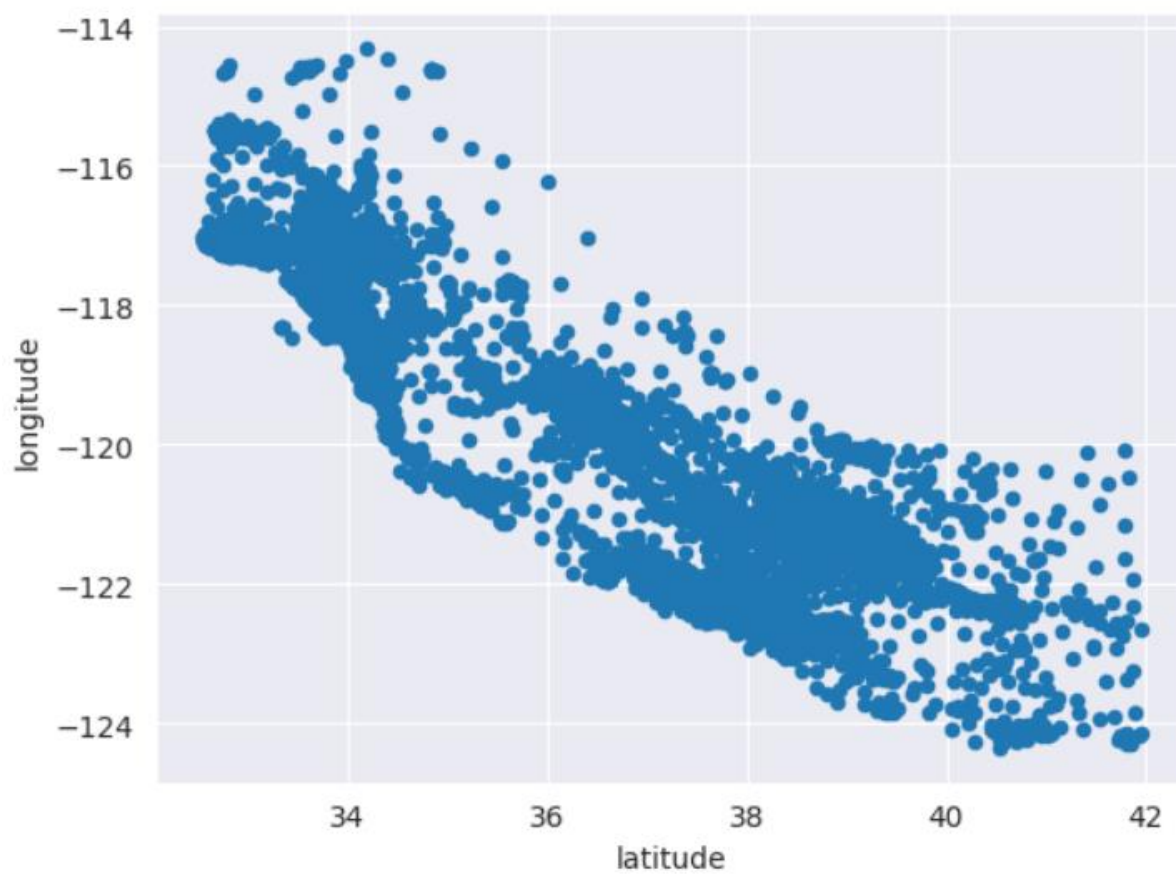


Hình 5. 7 Thu nhập bình quân của cư dân theo từng giá trị trong biến Ocean_proximity.

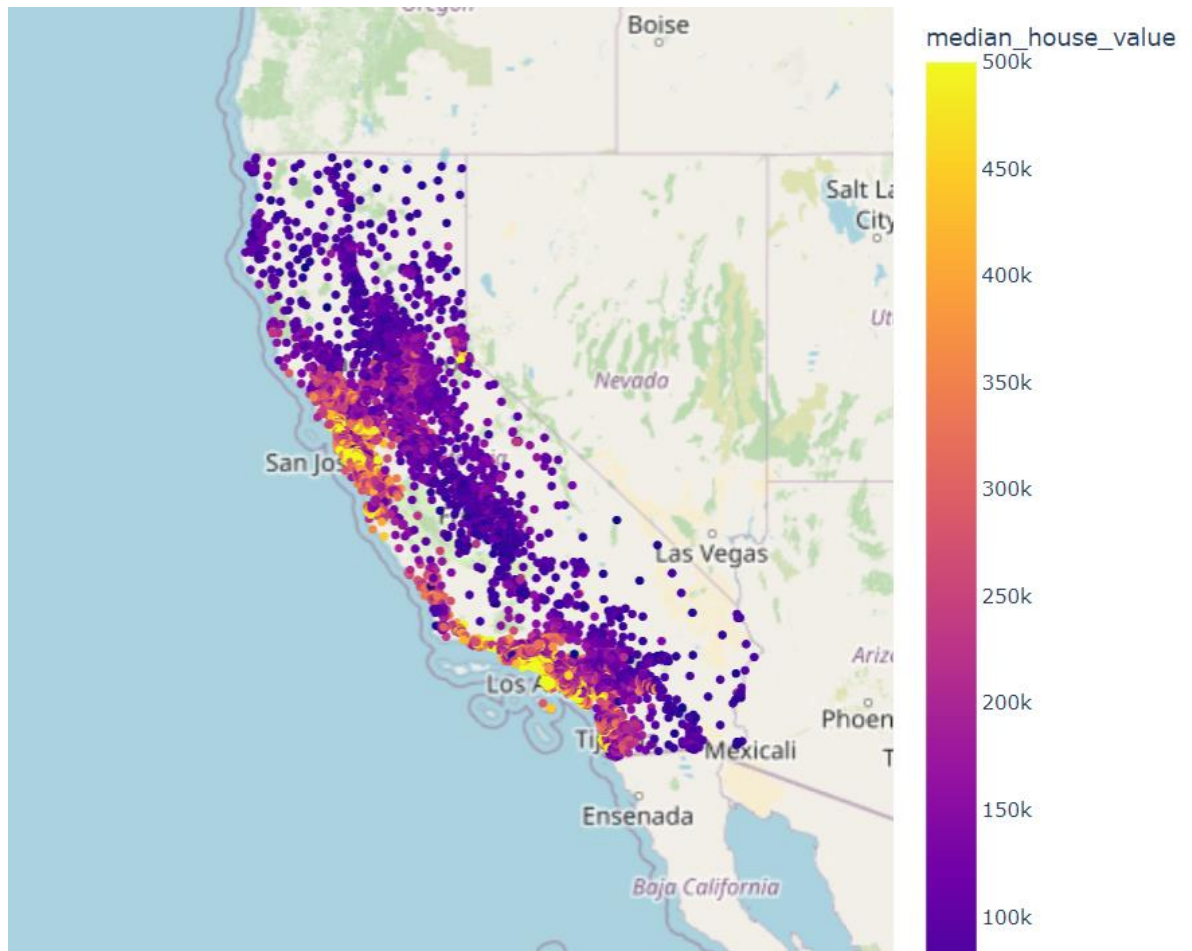
Tuy cư dân tập trung ở INLAND đông chiếm vị trí thứ 2, tuy nhiên thu nhập trung bình ở đây không cao. Ngược lại, NEAR OCEAN và NEAR BAY tuy dân cư không phân bố ở đây nhiều nhưng có thu nhập trung bình ở mức cao.

<1H OCEAN vẫn là nơi tập trung đông dân và có thu nhập đầu người cao. Đây là nơi có 2 đô thị vệ tinh như đã trình bày.

5.1.6 Price theo vị trí địa lý



Hình 5. 8 Sự phân bố theo vị trí địa lý (2 biến longitude và latitude)

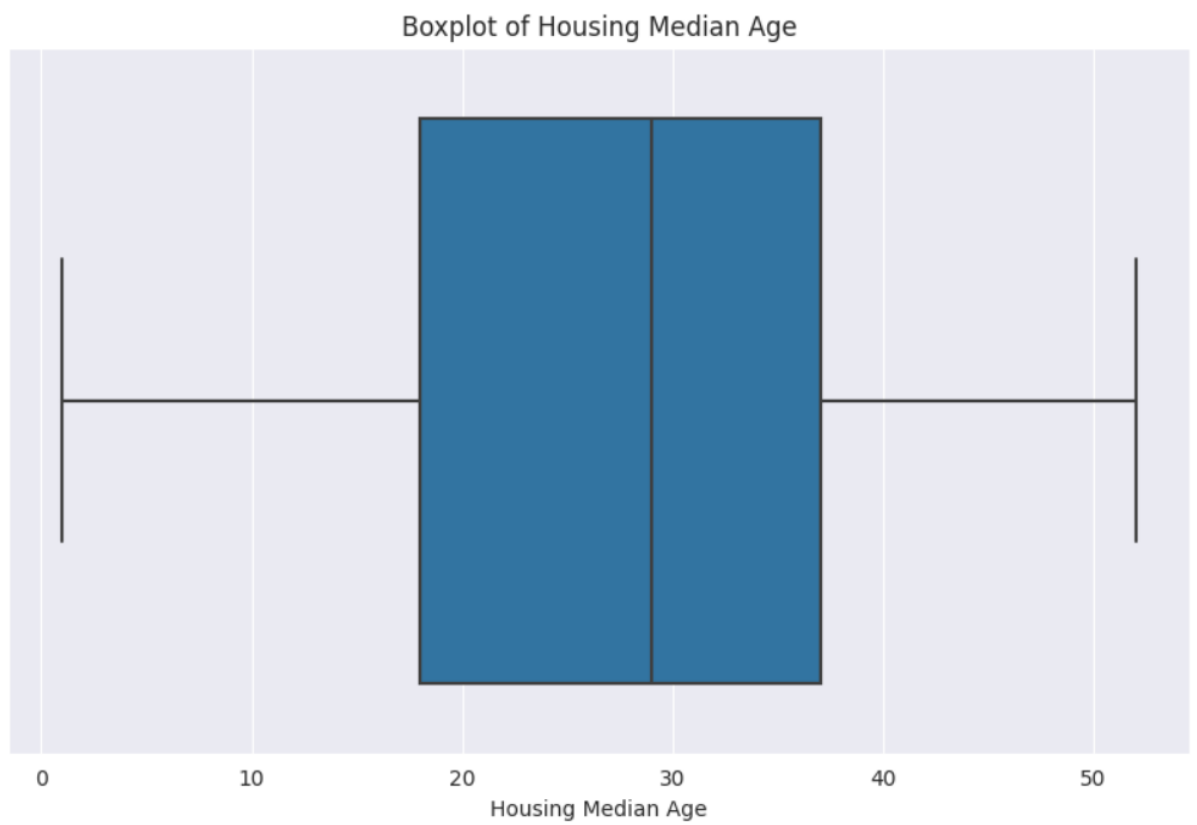


Hình 5. 9 Trực quan hóa trên bản đồ.

Mỗi một chấm tròn là tương ứng với một điểm dữ liệu với bán kính thể hiện số dân (cột population) và hiển thị màu sắc thay đổi theo giá nhà: màu xanh lá là giá nhà thấp và màu đỏ đậm là giá nhà cao. Đúng như quan sát ở trên, có 2 cụm dân cư lớn tập trung ở ven biển có mức giá nhà rất cao.

So sánh với sơ đồ thực tế, quan sát này là hợp lý. Khu vực Bay Area ở phía Bắc và Los Angeles ở phía nam thực sự là các khu dân cư lớn với giá bất động sản đắt đỏ.

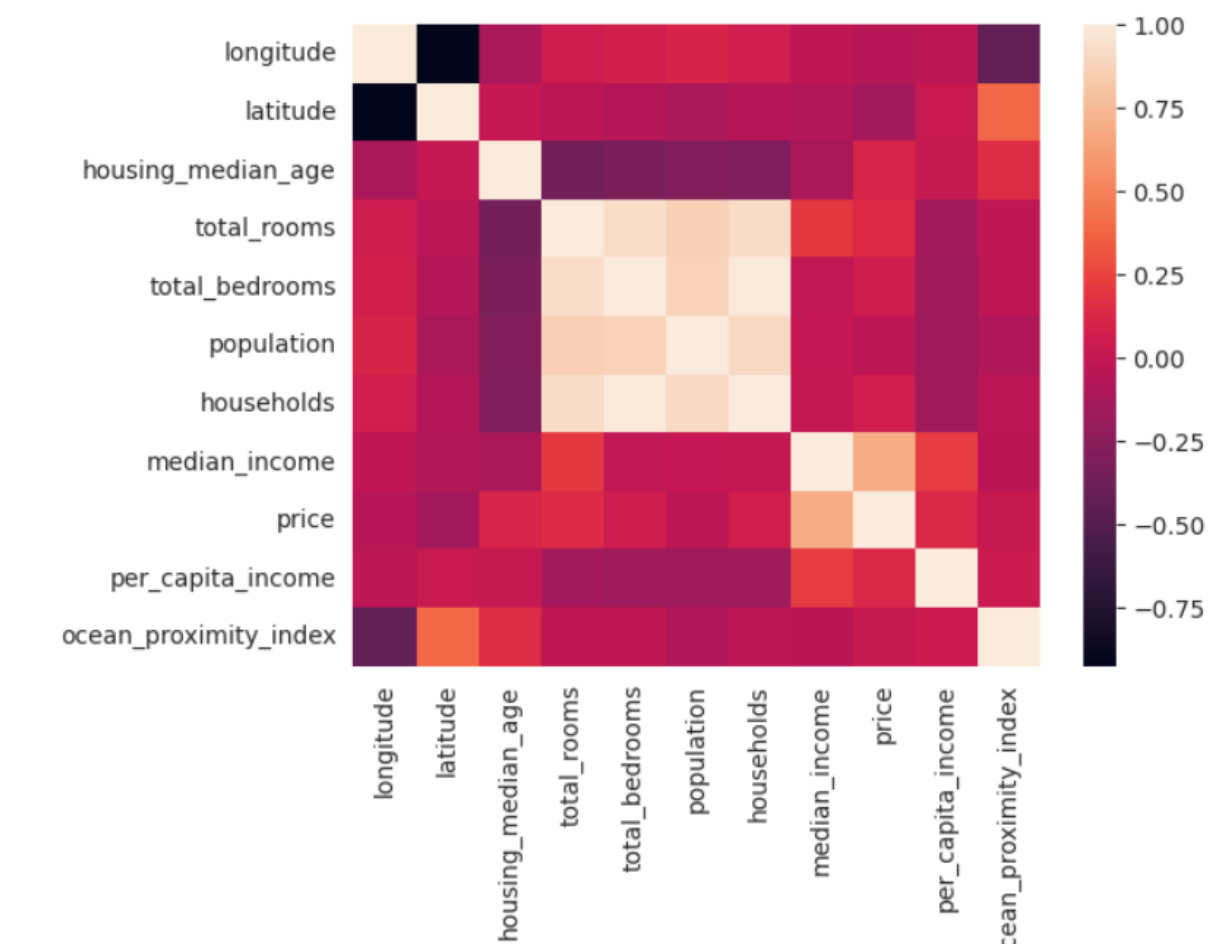
5.1.7 Phân tích biến Housing Median Age



Hình 5. 10 Sự phân bố của các giá trị trong biến `housing_median_age`

Giá trị số năm của các căn nhà trong khoảng từ 18-38 năm.

5.2 Ma trận tương quan



Hình 5. 11 Ma trận tương quan.

Biến mục tiêu price có mối tương quan rất nhẹ với tất cả ngoại trừ một đặc điểm ở đây: median_income, vì vậy người ta có thể coi đây là một đặc điểm quan trọng. Mối tương quan -0,02 & -0,05 (dân số/kinh độ) với biến mục tiêu Average_house_value có thể đáng để giảm, nhưng cũng có thể không. Trên thực tế, giá trị thấp không hẳn là lý do để loại bỏ một tính năng. Nó có thể chỉ đơn giản ngụ ý rằng dữ liệu được lan truyền khá nhiều, đây là một dấu hiệu mạnh mẽ về tính phi tuyến. Người ta thường khuyên nên bỏ các đặc điểm như vậy, đặc biệt đối với các mô hình ít phức tạp hơn, vì mô hình có thể sẽ không thể chọn ra một đặc điểm có tính phi tuyến như vậy chứ chưa nói đến nhiều đặc điểm

CHƯƠNG 6: THỰC NGHIỆM VÀ ĐÁNH GIÁ MÔ HÌNH

6.1 Feature Engineering

6.1.1 Thông tin các đặc trưng

```
root
|-- longitude: double (nullable = true)
|-- latitude: double (nullable = true)
|-- housing_median_age: double (nullable = true)
|-- total_rooms: double (nullable = true)
|-- total_bedrooms: double (nullable = true)
|-- population: double (nullable = true)
|-- households: double (nullable = true)
|-- median_income: double (nullable = true)
|-- median_house_value: double (nullable = true)
|-- ocean_proximity: string (nullable = true)
```

Hình 6. 1 Kiểm tra loại dữ liệu của các cột dữ liệu.

Nhìn chung bộ dữ liệu có nhiều trường mang kiểu dữ liệu “chuỗi”, do đó nhóm cần phải tiến hành chuyển đổi kiểu dữ liệu để phù hợp cho các bước xử lý và đưa vào mô hình

6.1.2 Encoding

```
#Label-encoding for the "ocean_proximity" column
from pyspark.ml.feature import StringIndexer
indexer = StringIndexer(inputCol="ocean_proximity", outputCol="ocean_proximity_index")
df_housing = indexer.fit(df_housing).transform(df_housing)
df_housing = df_housing.drop('ocean_proximity')
df_housing.select('ocean_proximity_index').show(3)
```

```
+-----+
|ocean_proximity_index|
+-----+
|                    3.0|
|                    3.0|
|                    3.0|
+-----+
only showing top 3 rows
```

Hình 6. 2 Chuyển đổi dữ liệu của cột “ocean_proximity” từ dạng chữ sang dạng số.

```

from pyspark.ml.feature import OneHotEncoder

one_hot_encoder = OneHotEncoder(inputCol='ocean_proximity_index',
                                outputCol='ocean_proximity_one_hot')

one_hot_encoder = one_hot_encoder.fit(train)

train = one_hot_encoder.transform(train)
test = one_hot_encoder.transform(test)

train.show(3)

```

```

+-----+-----+-----+-----+-----+-----+
|longitude|latitude|housing_median_age|total_rooms|total_bedrooms|pop|
+-----+-----+-----+-----+-----+-----+
|  -124.35|   40.54|           52.0|    1820.0|        300.0|   |
|  -124.3|   41.8|           19.0|    2672.0|        552.0|   |
|  -124.3|   41.84|           17.0|    2677.0|        531.0|   |
+-----+-----+-----+-----+-----+-----+
only showing top 3 rows

```

Hình 6. 3 Chuyển dữ liệu và gán chỉ mục cho các giá trị thuộc các cột.

Sử dụng thư viện PySpark để chuyển dữ liệu từ string sang numeric và gán chỉ mục được thực hiện bởi lệnh StringIndexer. Sau đó dùng OnehotEncoder để đưa các dữ liệu đó về dưới dạng vector để thuận lợi trong việc chạy mô hình.

Đầu ra của quá trình biến đổi trên là các biến có đuôi “one_hot”.

```

assembler = VectorAssembler(inputCols=['scaled_numerical_feature_vector',
                                         'ocean_proximity_one_hot'],
                             outputCol='final_feature_vector')

train = assembler.transform(train)
test = assembler.transform(test)

```

```

train.select('final_feature_vector').take(2)

```

```

[Row(final_feature_vector=DenseVector([-2.3829, 2.2717, 1.8876, -0.3749, -0.5743,
-0.5535, -0.6083, -0.423, 1.1094, 0.0, 0.0, 1.0, 0.0])),
 Row(final_feature_vector=DenseVector([-2.358, 2.857, -0.7452, 0.0186, 0.0237,
-0.1302, -0.066, -1.0887, 1.1094, 0.0, 0.0, 1.0, 0.0]))]

```

Hình 6. 4 Đưa dữ liệu về dạng vecto tổng hợp và phân tập train,test

Chúng ta có thể lưu mỗi một giá trị từ một cột như một phần tử của vector. Khi đó vector sẽ chứa toàn bộ các thông tin cần thiết của 1 quan sát để xác định nhãn hoặc giá trị dự báo ở đầu ra. Dùng lệnh “VectorAssembler” của `pyspark.ml.feature` sẽ tạo ra vector tổng hợp đại diện cho toàn bộ các chiều của quan sát đầu vào. Việc chúng ta cần thực hiện chỉ là truyền vào class list string tên các trường thành phần của vector tổng hợp.

6.2 Phân tập Train, Test và thực nghiệm các thuật toán

Ba thuật toán được lựa chọn để thực nghiệm mô hình là: Decision Tree, Random Forest, Linear regression.

```
# import libraries
from pyspark.ml.regression import LinearRegression
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
from pyspark.ml.regression import RandomForestRegressor
from pyspark.ml.regression import DecisionTreeRegressor

rf = RandomForestRegressor(featuresCol='final_feature_vector', labelCol='price')
lr = LinearRegression(featuresCol='final_feature_vector', labelCol='price')
dt = DecisionTreeRegressor(featuresCol='final_feature_vector', labelCol='price')
```

Hình 6. 5 Các thuật toán để thực nghiệm mô hình.

Biến phụ thuộc là giá vé “Price” và các biến độc lập đã được chuyển đổi thành Vector đặc trưng nhiều chiều. Tập dữ liệu được chia thành 2 tập Train và Test với tỷ lệ 80:20, tập Train dùng để huấn luyện mô hình, tập Test để kiểm tra và đánh giá tính chính xác của mô hình

6.3 Kết quả thực nghiệm và thảo luận

Để đánh giá các mô hình, nhóm lựa chọn các thông số MAE, MSE, RMSE, R2 Score:

- Mean Absolute Error (MAE): Giá trị sai số tuyệt đối trung bình (MAE) là thước đo cho thấy độ chính xác giữa các giá trị dự đoán so với giá trị thực tế. MAE tính trung bình của tổng các giá trị tuyệt đối các sai số.
- Mean Squared Error (MSE): Sai số bình phương trung bình (MSE) là giá trị trung bình của chênh lệch bình phương giữa giá trị dự đoán và giá trị quan sát được. MSE là thước đo chất lượng của mô hình hồi quy tuyến tính - nó luôn không âm và các giá trị càng gần 0 càng tốt.

- Root Mean Squared Error (RMSE): Sai số bình phương trung bình gốc (RMSE) là một thước đo được sử dụng để đánh giá sự khác biệt giữa các giá trị dự đoán và giá trị thực tế. RMSE được định nghĩa là căn bậc hai của sai số bình phương trung bình. RMSE càng nhỏ tức là sai số càng bé thì mức độ ước lượng cho thấy độ tin cậy của mô hình có thể đạt cao nhất.
- R2 Score

Model	MAE	MSE	RMSE	R2 Score
Random Forest	43949.813	3474008789.09	58940.72	0.614
Linear Regression	43532.06	3408877739.2	58385.59	0.622
Decision Tree	44873.91	3694386928.58	60781.468	0.59036

Từ kết quả thực nghiệm có thể thấy Linear Regression được đánh giá tốt hơn các thuật toán còn lại.

CHƯƠNG 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

7.1 Kết quả đạt được

- Việc hoàn thành thành công dự án này sẽ không chỉ cung cấp mô hình dự đoán về giá nhà đất ở các quận của California mà còn là ví dụ thực tế để nhóm tìm hiểu về học máy.
- Những hiểu biết sâu sắc thu được từ phân tích có thể góp phần hiểu rõ hơn về các yếu tố ảnh hưởng đến giá nhà đất, với những ứng dụng tiềm năng trong ngành bất động sản và quy hoạch đô thị

7.2 Hạn chế

- Mô hình chưa tối ưu, R2 Score còn thấp.
- Không có nhiều thời gian tìm hiểu và hiểu biết rộng về lĩnh vực bất động sản, vì vậy chưa khai thác sâu về các thông tin từ biểu đồ

7.3 Hướng phát triển

- Kết hợp nhiều thuật toán với nhau để bổ trợ và tận dụng nhiều hơn nữa các ưu điểm của từng thuật toán nhằm tăng thêm tính chính xác cũng như các chỉ số đánh giá được cao hơn.
- Thu thập thêm dữ liệu nhằm tăng tính khách quan, nhiều biến để quan sát toàn thể dữ liệu.
- Nghiên cứu thêm về thị trường bất động sản để có kiến thức chuyên sâu hơn về ngành.