# MỤC LỤC

# I. Giới thiệu vấn đề

## 1. Đặt vấn đề

Giá nhà là một thước đo quan trọng cho tình hình kinh tế của một quốc gia hay vùng lãnh thổ. Giá nhà tăng thường đi kèm với tăng trưởng kinh tế và gia tăng cơ hội việc làm [37], trong khi giá giảm có thể báo hiệu sự trì trệ kinh tế hoặc các yếu tố bất lợi khác. Trong những năm gần đây, sự phát triển nhanh chóng của ngành bất động sản đã đẩy giá nhà lên mức cao chưa từng thấy [1], thu hút sự chú ý của các cơ quan chính phủ, doanh nghiệp, cộng đồng và cá nhân.

Giữa bối cảnh bùng nổ bất động sản, xu hướng leo thang của giá nhà đã trở thành tâm điểm thảo luận. Tuy nhiên, những con số tăng trưởng ấn tượng đi kèm với những lo ngại ngày càng gia tăng, vì chúng phản ánh cả xu hướng vĩ mô và thực tế vi mô của nền kinh tế. Trong khi tăng trưởng nhanh có thể [4] tiềm ẩn rủi ro bong bóng, thì giá nhà tăng vọt đe dọa đến nhu cầu cơ bản về khả năng chi trả nhà ở. Các yếu tố ảnh hưởng đến giá nhà vượt xa các quy định vĩ mô. Thiết kế kiến trúc, cảnh quan, chất lượng xây dựng và thậm chí cả các yếu tố như chiếu sáng và bố trí đều có thể trực tiếp ảnh hưởng đến giá trị của từng bất động sản [5]. Chính sự đa dạng và biến động vốn có này góp phần tạo nên sự năng động của thị trường bất động sản, khiến việc điều hướng trở nên vừa thú vị vừa đầy thách thức.

Trong bối cảnh giá đất và nhà ở tăng hàng năm, việc dự đoán chính xác giá nhà đã trở thành một yêu cầu cấp thiết. Không chỉ người mua và người bán dựa vào ước tính chính xác để đưa ra quyết định sáng suốt [17], mà các nhà hoạch định chính sách cũng cần những dự đoán đáng tin cậy để định hướng thị trường ổn định và bền vững. Dự đoán giá nhà nổi lên như một vấn đề cốt lõi trong lĩnh vực bất động sản và kinh tế, đòi hỏi nghiên cứu chuyên sâu và phát triển các phương pháp tiên tiến có thể góp phần vào sự ổn định của thị trường và bảo vệ lợi ích của tất cả các bên liên quan.

**2. Giới thiệu**

Giá nhà luôn là chủ đề thu hút sự quan tâm của nhiều người, đặc biệt là trong bối cảnh thị trường bất động sản biến động liên tục. Việc dự đoán giá bán chính xác cho từng ngôi nhà có thể mang lại nhiều lợi ích cho các nhà đầu tư, môi giới và người mua nhà., mục tiêu của việc nghiên cứu là tập trung vào dự đoán giá bán cho từng ngôi nhà ở Ames , một thành phố thuộc Quận Story trong bang Iowa, Hoa Kỳ. Thành phố có tổng diện tích 62.86 km², trong đó diện tích đất là 62.70 km².

Các thuật toán phổ biến trong việc dự đoán giá nhà như : Hồi quy tuyến tính , Random Forest , Gradient Boosted Trees , … Dữ liệu nghiên cứu là một tập dữ liệu Với 79 biến giải thích mô tả (gần như) mọi khía cạnh của nhà ở Ames, Iowa . Một số trường dữ liệu tiêu biểu như : Sale Price , Electrical , Garage Area , Sale Type , Functional.

Kết quả của nghiên cứu này sẽ giúp các nhà đầu tư, môi giới và người mua nhà có thể đưa ra quyết định sáng suốt hơn trong việc mua bán nhà ở Ames. Mặc dù nghiên cứu chỉ tập trung vào dự đoán giá nhà ở Ames, Iowa, Hoa Kỳ , nhưng những phương pháp và kết quả cũng có thể đóng góp vào việc phát triển các mô hình học máy hiệu quả hơn để dự đoán giá nhà ở các khu vực khác.

## II. Các nghiên cứu liên quan

Trước thách thức khó khăn trong việc thị trường bất động chuyển biến bất thường, một loạt các nghiên cứu đã ra đời, mỗi nghiên cứu đều tìm cách khám phá những bí mật ẩn chứa trong lĩnh vực dự đoán giá nhà. Từ các kỹ thuật hồi quy tiên tiến được khám phá trong "Dự đoán Giá Nhà Sử Dụng Các Kỹ Thuật Hồi Quy Tiên Tiến [1]" đến phương pháp hồi quy có trọng số theo biến phụ thuộc được championed (bị ủng hộ) trong "Hồi Quy Có Trọng Số Theo Biến Phụ Thuộc (CWR): Nghiên cứu Trường hợp để Ước Tính Giá Nhà [9]", hồi quy theo trọng số địa lý (GWR) là một công cụ phổ biến để mô hình hóa tính không đồng nhất về mặt không gian trong một mô hình hồi quy. Tuy nhiên, hàm trọng số hiện tại được sử dụng trong GWR chỉ xét đến khoảng cách địa lý, trong khi sự tương đồng về thuộc tính hoàn toàn bị bỏ qua. Trong nghiên cứu này, chúng tôi đề xuất một hàm trọng số theo biến phụ thuộc kết hợp cả khoảng cách địa lý và khoảng

cách thuộc tính. Hồi quy theo trọng số khoảng cách biến phụ thuộc (CWR) là sự mở rộng của GWR bao gồm khoảng cách địa lý và khoảng cách thuộc tính. Giá nhà bị ảnh hưởng bởi nhiều yếu tố, chẳng hạn như tuổi nhà, diện tích sàn và mục đích sử dụng đất. Mô hình dự đoán được sử dụng để giúp hiểu các đặc điểm của giá nhà theo vùng. CWR được sử dụng để hiểu mối quan hệ giữa giá nhà và các yếu tố kiểm soát. CWR có thể xem xét khoảng cách địa chất và khoảng cách thuộc tính, và đưa ra ước tính chính xác về giá nhà, bảo toàn ma trận trọng số cho các hàm khoảng cách địa chất và khoảng cách thuộc tính. Kết quả cho thấy các thuộc tính / điều kiện của nhà và các đặc điểm của ngôi nhà, chẳng hạn như diện tích sàn và tuổi nhà, có thể ảnh hưởng đến giá nhà. Sau khi lựa chọn yếu tố, trong đó chỉ xem xét tuổi nhà và diện tích sàn của tòa nhà, RMSE của mô hình CWR có thể được cải thiện từ 2,9% - 26,3% đối với các tòa nhà chọc trời so với GWR. CWR có thể giảm hiệu quả các lỗi ước tính từ các mô hình hồi quy không gian truyền thống và cung cấp các mô hình mới và khả thi cho ước tính không gian .

Các nhà nghiên cứu còn sử dụng các mô hình kết hợp để dự đoán giá nhà Một mô hình pha trộn dựa trên Python và các gói xgboost, DF21 và Geatpy của nó để dự đoán giá nhà bán lại ở Singapore [ 11 ]. Đầu tiên, các thuộc tính phân loại hồng y cao được xử lý trước bằng phương pháp mã hóa trung bình. Sau đó, các nhà nghiên cứu  đề xuất một phương pháp pha trộn tuyến tính bao gồm GA-HL-Xg-Boost, GARandom Forest (GA-RF), deep-Random Forest (DRF) và lightGBM, với tạp chất Gini để xác định tầm quan trọng của các tính năng. Cuối cùng, kết quả cho thấy nó có thể đạt được sai số phần trăm tuyệt đối trung bình (MAPE) là 7,36% trong xu hướng chung cố định của xu hướng giá nhà. Nghiên cứu này có thể cung cấp một dự báo mạnh mẽ về giá bán lại nhà trong các môi trường kinh tế khác nhau.

Ngoài ra , có một vài công trình báo cáo việc sử dụng thuật toán học máy (ML) để dự đoán giá nhà . Tại Brazill , có một nghiên cứu này phân tích một bộ dữ liệu bao gồm 12.223.582 quảng cáo nhà ở, được thu thập từ các trang web của Brazil từ năm 2015 đến 2018. Mỗi phiên bản bao gồm hai mươi bốn tính năng của năm loại dữ liệu khác nhau: số nguyên, ngày, chuỗi, float và hình ảnh. Để dự đoán giá bất động sản, họ kết hợp hai kiến trúc ML khác nhau, dựa trên [ 15 ] Random Forest (RF) và Recurrent Neural Networks (RNN). Nghiên cứu này chứng minh rằng làm phong phú bộ dữ liệu

và kết hợp các phương pháp ML khác nhau có thể là một giải pháp thay thế tốt hơn để dự đoán giá nhà ở những khu vực khác nhau

# III. Phương pháp

## 3.1. Kỹ thuật nền tảng

### 3.1.1. Sai số trung bình bình phương (MSE)

Sai số trung bình bình phương (MSE) là thước đo thống kê được sử dụng để đánh giá mức độ chênh lệch giữa giá trị dự đoán và giá trị thực tế trong các mô hình học máy.

Được minh hoạ như sau: $$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}( Y_i - \hat{Y}_i)^2$$

Trong đó: n là số lượng dữ liệu

Yi là giá trị thực tế

Y^i là giá trị dự đoán

Giá trị MSE càng thấp, mô hình càng chính xác hơn. Giá trị MSE càng cao, mô hình càng kém chính xác. MSE được sử dụng rộng rãi để đánh giá hiệu quả của các mô hình học máy trong nhiều lĩnh vực, bao gồm: Đánh giá mức độ chính xác của mô hình trong việc dự đoán giá trị liên tục ( Hồi quy ) và Đánh giá mức độ chính xác của mô hình trong việc phân loại các mẫu dữ liệu ( Phân loại )

MSE được đánh giá dễ hiểu và tính toán đơn giản, có thể so sánh được giữa các mô hình khác nhau. Tuy nhiên mặt hạn chế là nhạy cảm với các giá trị ngoại lai, các giá trị ngoại lai có thể làm tăng giá trị MSE và ảnh hưởng đến đánh giá của mô hình. Trong dự đoán giá bán nhà, MSE được sử dụng để đánh giá mức độ chính xác của mô hình Random Forest trong việc dự đoán giá bán nhà. Giá trị MSE càng thấp, mô hình càng dự đoán chính xác giá bán nhà.

### 3.1.2. Căn bậc hai trung bình bình phương sai số (RMSE)

Căn bậc hai trung bình bình phương sai số (RMSE) là thước đo thống kê được sử dụng để đánh giá mức độ chênh lệch giữa giá trị dự đoán và giá trị thực tế trong các mô hình học máy.

RMSE là căn bậc hai của MSE (Sai số Trung bình Bình phương) có công thức như sau:
**RMSE = √(MSE)**

RMSE có cùng đơn vị đo lường với giá trị thực tế, giúp dễ dàng đánh giá mức độ sai lệch. Giá trị RMSE càng thấp, mô hình càng chính xác hơn. Giá trị RMSE càng cao, mô hình càng kém chính xác.RMSE được sử dụng rộng rãi để đánh giá hiệu quả của các mô hình học máy trong nhiều lĩnh vực, bao gồm: Đánh giá mức độ chính xác của mô hình trong việc dự đoán giá trị liên tục ( Hồi quy ) và Đánh giá mức độ chính xác của mô hình trong việc phân loại các mẫu dữ liệu ( Phân loại )

RMSE được đánh giá dễ hiểu và tính toán đơn giản. Có đơn vị đo lường, giúp dễ dàng so sánh mức độ sai lệch giữa các mô hình. Tuy nhiên mặt hạn chế là nhạy cảm với các giá trị ngoại lai và RMSE không có ý nghĩa thống kê.Trong dự đoán giá bán nhà, RMSE được sử dụng để đánh giá mức độ chính xác của mô hình Random Forest trong việc dự đoán giá nhà. Giá trị RMSE càng thấp, mô hình càng dự đoán chính xác giá bán nhà.

### 3.1.3. R^2: Hệ số tương quan xác định

R^2 (Hệ số tương quan xác định) là thước đo thống kê được dùng để đánh giá mức độ tương quan giữa giá trị dự đoán và giá trị thực tế trong các mô hình học máy. R^2 thể hiện tỷ lệ phần trăm biến thiên của biến phụ thuộc được giải thích bởi các biến độc lập.

Công thức**:** **R^2 = 1 - (SSR / SST)**

Trong đó : R^2: Hệ số tương quan xác định

SSR: Sum of Squared Residuals (Tổng Bình phương Sai số).

SST: Sum of Squared Totals (Tổng Bình phương Toàn phần)

Giá trị R^2 càng cao, mô hình càng có khả năng dự đoán tốt hơn. Giá trị R^2 càng thấp, mô hình càng có khả năng dự đoán kém hơn. Giá trị R^2 không âm và không thể lớn hơn 1. R^2 được sử dụng rộng rãi để đánh giá hiệu quả của các mô hình học máy trong nhiều lĩnh vực, bao gồm: Đánh giá mức độ chính xác của mô hình trong việc dự đoán giá trị liên tục ( Hồi quy ) và Đánh giá mức độ chính xác của mô hình trong việc phân loại các mẫu dữ liệu ( Phân loại )

R^2 được đánh giá dễ hiểu và tính toán đơn giản, có thể so sánh được giữa các mô hình khác nhau. Tuy nhiên mặt hạn chế là không có ý nghĩa thống kê, nhạy cảm với các giá trị ngoại lai.

Trong dự đoán giá bán nhà, R^2 được sử dụng để đánh giá mức độ chính xác của mô hình Random Forest trong việc dự đoán giá bán nhà. Giá trị R^2 càng cao, mô hình càng dự đoán chính xác giá bán nhà.

### 3.1.4. Adjusted R^2: Hệ số tương quan xác định điều chỉnh

Adjusted R^2 là phiên bản điều chỉnh của R^2, giúp bù đắp cho việc số lượng biến dự đoán ảnh hưởng đến giá trị R^2. Adjusted R^2 có xu hướng giảm khi thêm nhiều biến dự đoán không quan trọng vào mô hình.

Công thức : **Adjusted R^2 = 1 - [(1 - R^2) * (n - 1) / (n - k)]**

Trong đó : Adjusted R^2: Hệ số tương quan xác định điều chỉnh

R^2: Hệ số tương quan xác định

n : Số lượng dữ liệu

k : Số lượng biến dự đoán

Giá trị Adjusted R^2 càng cao, mô hình càng có khả năng dự đoán tốt hơn. Giá trị Adjusted R^2 càng thấp, mô hình càng có khả năng dự đoán kém hơn. Giá trị Adjusted R^2 không âm và không thể lớn hơn 1. Adjusted R^2 được sử dụng rộng rãi để đánh giá hiệu quả của các mô hình học máy trong nhiều lĩnh vực, bao gồm: Đánh giá mức độ

chính xác của mô hình trong việc dự đoán giá trị liên tục (Hồi quy) và Đánh giá mức độ chính xác của mô hình trong việc phân loại các mẫu dữ liệu (Phân loại)

Adjusted R^2 giúp so sánh hiệu quả giữa các mô hình có số lượng biến dự đoán khác nhau, bù đắp cho việc R^2 có xu hướng tăng khi thêm nhiều biến dự đoán. Tuy nhiên nó không có ý nghĩa thống kê và nhạy cảm với các giá trị ngoại lai. Trong dự đoán giá bán nhà, Adjusted R^2 được sử dụng để đánh giá mức độ chính xác của mô hình Random Forest trong việc dự đoán giá bán nhà. Giá trị Adjusted R^2 càng cao, mô hình càng dự đoán chính xác giá bán nhà. Cần so sánh giá trị Adjusted R^2 của mô hình Random Forest với các mô hình khác để đánh giá hiệu quả của mô hình.Cả R^2 và Adjusted R^2 đều là thước đo thống kê, do đó không nên sử dụng chúng để đưa ra quyết định cuối cùng về việc lựa chọn mô hình.

### 3.1.5. Chỉ số MAE (Mean Absolute Error)

Chỉ số MAE (Mean Absolute Error) là thước đo thống kê được sử dụng để đánh giá mức độ lệch trung bình giữa giá trị dự đoán và giá trị thực tế trong các mô hình học máy. MAE là giá trị trung bình của các giá trị tuyệt đối của sai số.

Công thức: $$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Trong đó : MAE: Mean Absolute Error (Chỉ số Sai số Tuyệt đối Trung bình)

n: Số lượng dữ liệu

Yi: Giá trị thực tế

Xi: Giá trị dự đoán

Giá trị MAE càng thấp, mô hình càng chính xác hơn. Giá trị MAE càng cao, mô hình càng kém chính xác.MAE được sử dụng rộng rãi để đánh giá hiệu quả của các mô hình học máy trong nhiều lĩnh vực, bao gồm Đánh giá mức độ chính xác của mô hình trong

việc dự đoán giá trị liên tục (Hồi quy) và Đánh giá mức độ chính xác của mô hình trong việc phân loại các mẫu dữ liệu (Phân loại).

MAE được đánh giá dễ hiểu và tính toán đơn giản, có đơn vị đo lường và it nhạy cảm với các giá trị ngoại lai hơn so với MSE. Tuy nhiên, MAE không có ý nghĩa thống kê và không phạt sai số lớn như MSE. Trong dự đoán giá bán nhà, MAE được sử dụng để đánh giá mức độ chính xác của mô hình Random Forest trong việc dự đoán giá bán nhà. Giá trị MAE càng thấp, mô hình càng dự đoán chính xác giá bán nhà. Tuy nhiên, cần so sánh giá trị MAE của mô hình Random Forest với các mô hình khác để đánh giá hiệu quả của mô hình. Cả MAE và MSE đều là thước đo thống kê, do đó không nên sử dụng chúng để đưa ra quyết định cuối cùng về việc lựa chọn mô hình. Cần xem xét các yếu tố khác như độ phức tạp của mô hình, khả năng giải thích và khả năng tổng quát hóa khi lựa chọn mô hình.

### 3.1.6. Chỉ số MAPE (Mean Absolute Percentage Error)

Chỉ số MAPE (Mean Absolute Percentage Error) là thước đo thống kê được sử dụng để đánh giá mức độ lệch trung bình phần trăm giữa giá trị dự đoán và giá trị thực tế trong các mô hình học máy. MAPE thể hiện sai số trung bình theo tỷ lệ phần trăm.

Công thức:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

Trong đó : Percentage Error (Chỉ số Sai số Tuyệt đối Trung bình Phần trăm)

n: Số lượng dữ liệu

At: Giá trị thực tế

Ft: Giá trị dự đoán

Giá trị MAPE càng thấp, mô hình càng chính xác hơn. Giá trị MAPE càng cao, mô hình càng kém chính xác. MAPE có thể được so sánh trực tiếp giữa các mô hình dự đoán giá trị phần trăm. MAPE được sử dụng rộng rãi để đánh giá hiệu quả của các mô hình học

máy trong các lĩnh vực: Đánh giá mức độ chính xác của mô hình trong việc dự đoán giá trị phần trăm ( Hồi quy) và Đánh giá mức độ chính xác của mô hình trong việc phân loại các mẫu dữ liệu (Phân loại).

Mô hình MAPE được đánh giá dễ hiểu và tính toán đơn giản. Có đơn vị đo lường (phần trăm) và cho phép so sánh trực tiếp giữa các mô hình dự đoán giá trị phần trăm. Tuy nhiên, MAPE không có ý nghĩa thống kê, nhạy cảm với các giá trị 0 hoặc gần 0 trong giá trị thực tế và phạt sai số nhỏ nhiều hơn so với sai số lớn. Trong dự đoán giá bán nhà, MAPE được sử dụng để đánh giá mức độ chính xác của mô hình Random Forest trong việc dự đoán giá bán nhà theo tỷ lệ phần trăm. Giá trị MAPE càng thấp, mô hình càng dự đoán chính xác giá bán nhà.

### 3.2 Đề xuất phương pháp

TensorFlow Decision Forests (TF-DF) là một thư viện mã nguồn mở được phát triển bởi Google, cung cấp các công cụ mạnh mẽ để xây dựng và triển khai mô hình rừng quyết định (decision forest) trên nền tảng TensorFlow. TF-DF mang đến nhiều lợi ích cho việc phát triển mô hình học máy, bao gồm:

- Hiệu suất cao: TF-DF được tối ưu hóa cho hiệu suất cao, giúp tăng tốc độ đào tạo và dự đoán mô hình.
- Khả năng mở rộng: TF-DF có thể xử lý lượng dữ liệu lớn một cách hiệu quả, phù hợp cho các ứng dụng học máy quy mô lớn.
- Dễ sử dụng: TF-DF cung cấp API đơn giản và dễ sử dụng, giúp bạn dễ dàng xây dựng và triển khai mô hình rừng quyết định.
- Tính linh hoạt: TF-DF hỗ trợ nhiều loại mô hình rừng quyết định khác nhau, cho phép bạn lựa chọn mô hình phù hợp nhất với nhu cầu của mình.

TF-DF bao gồm các thành phần chính sau:

- Bộ xây dựng mô hình: Cung cấp các công cụ để xây dựng các loại mô hình rừng quyết định khác nhau, bao gồm Random Forest, Gradient Boosting Trees, và XGBoost.

- Bộ xử lý dữ liệu: Cung cấp các công cụ để chuẩn bị dữ liệu cho việc đào tạo mô hình, bao gồm xử lý thiếu dữ liệu, mã hóa dữ liệu, và chia tập dữ liệu.
- Bộ đánh giá mô hình: Cung cấp các công cụ để đánh giá hiệu suất mô hình, bao gồm tính toán các chỉ số đánh giá như độ chính xác, độ F1, và AUC.
- Bộ triển khai mô hình: Cung cấp các công cụ để triển khai mô hình vào sản xuất, bao gồm lưu mô hình, xuất mô hình sang TensorFlow Lite, và sử dụng mô hình trong các ứng dụng web và di động.

TF-DF là một thư viện mạnh mẽ và dễ sử dụng cho việc xây dựng và triển khai mô hình rừng quyết định trên nền tảng TensorFlow. TF-DF cung cấp nhiều tính năng và lợi ích giúp phát triển mô hình dự đoán giá nhà.

Hồi quy tuyến tính là một mô hình thống kê được sử dụng để dự đoán giá trị của một biến phụ thuộc dựa trên một hoặc nhiều biến độc lập. Mô hình sử dụng phương trình tuyến tính để mô tả mối quan hệ giữa các biến.

- Đơn giản và dễ hiểu: Hồi quy tuyến tính là một mô hình học máy đơn giản và dễ hiểu. Mô hình này sử dụng phương trình tuyến tính để dự đoán giá nhà dựa trên các biến độc lập. Nhờ vậy, việc triển khai và giải thích mô hình trở nên dễ dàng hơn so với các mô hình học máy phức tạp khác.
- Hiệu quả: Hồi quy tuyến tính là một mô hình học máy hiệu quả. Mô hình này có thể học nhanh và dự đoán chính xác với lượng dữ liệu nhỏ. Điều này giúp tiết kiệm thời gian và chi phí khi triển khai mô hình.
- Khả năng giải thích: Hồi quy tuyến tính cung cấp khả năng giải thích cao. Mô hình này cho phép xác định được mức độ ảnh hưởng của từng biến độc lập đến giá nhà. Nhờ vậy, người dùng có thể đưa ra quyết định đầu tư sáng suốt hơn.
- Khả năng mở rộng: Hồi quy tuyến tính có thể dễ dàng mở rộng để xử lý lượng dữ liệu lớn. Điều này giúp mô hình phù hợp với việc dự đoán giá nhà trong các thị trường bất động sản lớn.
- Tốc độ xử lý nhanh: Hồi quy tuyến tính có tốc độ xử lý nhanh hơn so với các mô hình học máy phức tạp khác. Điều này giúp mô hình phù hợp với việc dự đoán giá nhà trong thời gian thực.

Sử dụng Hồi quy tuyến tính để dự đoán giá nhà có nhiều ưu điểm, bao gồm đơn giản, hiệu quả, khả năng giải thích, khả năng mở rộng, tốc độ xử lý nhanh. Do đó, Hồi quy tuyến tính là một lựa chọn tốt cho việc dự đoán giá nhà trong thực tế, đặc biệt là khi dữ liệu có sẵn là tuyến tính.

# IV. Thực nghiệm

## 4.1. Miêu tả dữ liệu

### 4.1.1. Tổng quan

Nguồn dữ liệu: Dữ liệu được cung cấp bởi giảng viên bộ môn

### 4.1.2. Đặc điểm dữ liệu

Nhập thư viện

In [1]:
```
import tensorflow as tf
import tensorflow_decision_forests as tfdf
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Comment this if the data visualisations doesn't work on your side
%matplotlib inline
```

In [2]:
```
print("TensorFlow v" + tf.__version__)
print("TensorFlow Decision Forests v" + tfdf.__version__)
```

```
TensorFlow v2.11.0
TensorFlow Decision Forests v1.2.0
```

Tải dữ liệu lên

In [3]:
```
train_file_path = "../input/house-prices-advanced-regression-techniques/train.csv"
dataset_df = pd.read_csv(train_file_path)
print("Full train dataset shape is {}".format(dataset_df.shape))
```

```
Full train dataset shape is (1460, 81)
```

In [6]:
```
dataset_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 80 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   MSSubClass     1460 non-null   int64  
 1   MSZoning       1460 non-null   object 
 2   LotFrontage    1201 non-null   float64
 3   LotArea        1460 non-null   int64  
 4   Street         1460 non-null   object 
 5   Alley          91 non-null     object 
 6   LotShape       1460 non-null   object 
 7   LandContour    1460 non-null   object 
 8   Utilities      1460 non-null   object 
 9   LotConfig      1460 non-null   object 
 10  LandSlope      1460 non-null   object 
 11  Neighborhood   1460 non-null   object 
 12  Condition1     1460 non-null   object 

 78  SaleCondition  1460 non-null   object 
 79  SalePrice      1460 non-null   int64  
dtypes: float64(3), int64(34), object(43)
memory usage: 912.6+ KB
```

Dữ liệu chứa 80 thuộc tính và 1459 bản ghi

In [7]:
```
print(dataset_df['SalePrice'].describe())
plt.figure(figsize=(9, 8))
sns.distplot(dataset_df['SalePrice'], color='g', bins=100, hist_kws={'alpha': 0.4});
```

```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

Xem xét cách các đặc trưng số học được phân phối. Để làm điều này, trước tiên hãy liệt kê tất cả các loại dữ liệu từ tập dữ liệu và chỉ chọn các loại dữ liệu số học

In [8]:
```
list(set(dataset_df.dtypes.tolist()))
```

Out[8]:
```
[dtype('O'), dtype('int64'), dtype('float64')]
```

Có ba loại dữ liệu trong tập dữ liệu của chúng ta:

- dtype('O'): Dữ liệu đối tượng (object), thường là các chuỗi ký tự.
- dtype('int64'): Dữ liệu số nguyên 64-bit.
- dtype('float64'): Dữ liệu số thực 64-bit.

Chúng ta sẽ chỉ chọn các đặc trưng có kiểu dữ liệu số học (int64 và float64) để phân tích phân phối.

In [9]:
```
df_num = dataset_df.select_dtypes(include = ['float64', 'int64'])
df_num.head()
```

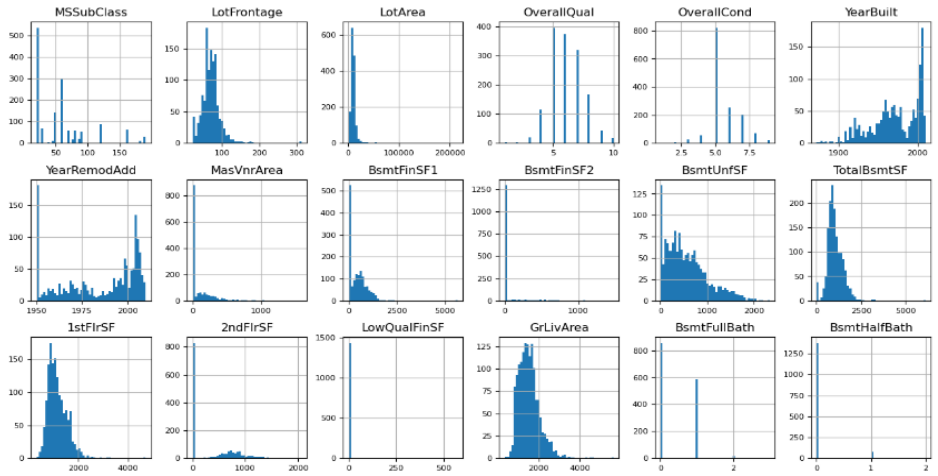
Out[9]:

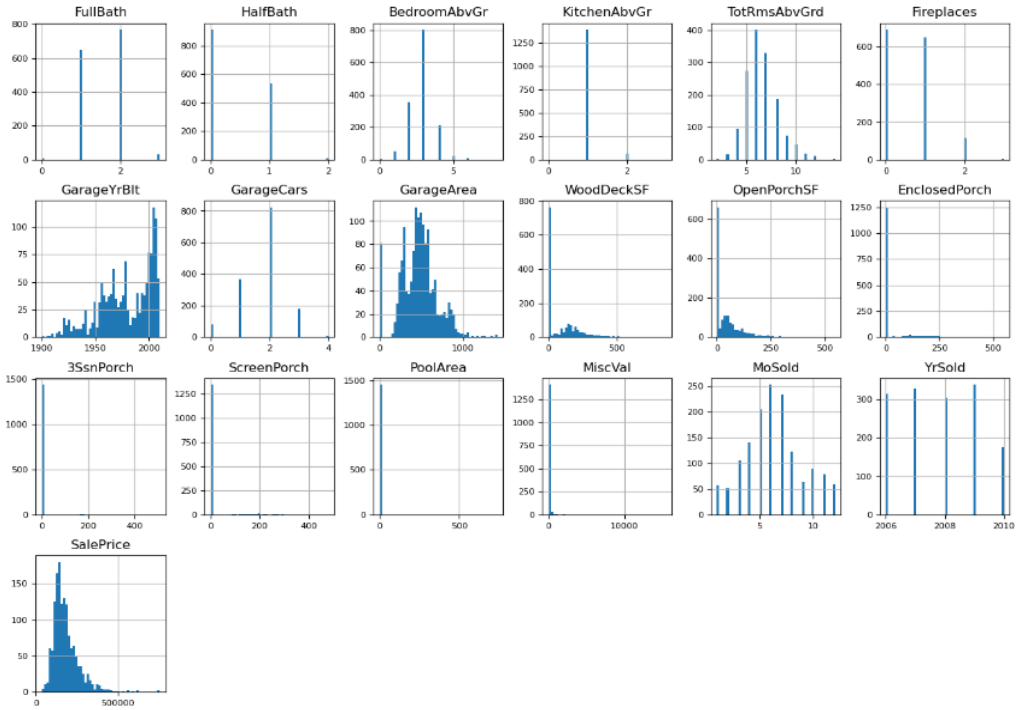
| | MSSubClass | LotFrontage | LotArea | OverallQual | OverallCond | YearBuilt | YearRemodAdd | M |
|---|---|---|---|---|---|---|---|---|
| 0 | 60 | 65.0 | 8450 | 7 | 5 | 2003 | 2003 | 1 |
| 1 | 20 | 80.0 | 9600 | 6 | 8 | 1976 | 1976 | 0 |
| 2 | 60 | 68.0 | 11250 | 7 | 5 | 2001 | 2002 | 1 |
| 3 | 70 | 60.0 | 9550 | 7 | 5 | 1915 | 1970 | 0 |
| 4 | 60 | 84.0 | 14260 | 8 | 5 | 2000 | 2000 | 3 |

5 rows × 37 columns

Bây giờ chúng ta hãy vẽ biểu đồ phân phối cho tất cả các đặc điểm số.

In [10]:
```
df_num.hist(figsize=(16, 20), bins=50, xlabelsize=8, ylabelsize=8);
```






```
# Correlation matrix of numerical features
corr_matrix = train_df.select_dtypes(include=['int64', 'float64']).corr()

# Heatmap of correlations
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix, cmap='coolwarm', annot=False, linewidths=.5)
plt.title('Correlation Heatmap')
plt.show()
```

In [11]:

```
import numpy as np

def split_dataset(dataset, test_ratio=0.30):
  test_indices = np.random.rand(len(dataset)) < test_ratio
  return dataset[~test_indices], dataset[test_indices]

train_ds_pd, valid_ds_pd = split_dataset(dataset_df)
print("{} examples in training, {} examples in testing.".format(
    len(train_ds_pd), len(valid_ds_pd)))
```

```
1010 examples in training, 450 examples in testing.
```

Danh sách các mô hình TensorFlow Decision Forests:

- tensorflow_decision_forests.keras.RandomForestModel: Mô hình rừng ngẫu nhiên.
- tensorflow_decision_forests.keras.GradientBoostedTreesModel: Mô hình máy tăng cường độ Gradient.
- tensorflow_decision_forests.keras.CartModel: Mô hình cây quyết định.
- tensorflow_decision_forests.keras.DistributedGradientBoostedTreesModel: Mô hình máy tăng cường độ Gradient phân tán.

In [12]:
```
label = 'SalePrice'
train_ds = tfdf.keras.pd_dataframe_to_tf_dataset(train_ds_pd, label=labe
l, task = tfdf.keras.Task.REGRESSION)
valid_ds = tfdf.keras.pd_dataframe_to_tf_dataset(valid_ds_pd, label=labe
l, task = tfdf.keras.Task.REGRESSION)
```

In [13]:
```
tfdf.keras.get_all_models()
```

Out[13]:
```
[tensorflow_decision_forests.keras.RandomForestModel,
 tensorflow_decision_forests.keras.GradientBoostedTreesModel,
 tensorflow_decision_forests.keras.CartModel,
 tensorflow_decision_forests.keras.DistributedGradientBoostedTreesMod
el]
```

### 4.2. Tiền xử lí dữ liệu

```
df.head()
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 80 columns):
Id              1459 non-null int64
MSSubClass      1459 non-null int64
MSZoning        1455 non-null object
LotFrontage     1232 non-null float64
```

```
df.describe()
```

| | Id | MSSubClass | LotFrontage | LotArea | OverallQual | OverallCond | Yearl |
|---|---|---|---|---|---|---|---|
| **count** | 1459.000000 | 1459.000000 | 1232.000000 | 1459.000000 | 1459.000000 | 1459.000000 | 1459.00( |
| **mean** | 2190.000000 | 57.378341 | 68.580357 | 9819.161069 | 6.078821 | 5.553804 | 1971.35 |
| **std** | 421.321334 | 42.746880 | 22.376841 | 4955.517327 | 1.436812 | 1.113740 | 30.39( |
| **min** | 1461.000000 | 20.000000 | 21.000000 | 1470.000000 | 1.000000 | 1.000000 | 1879.00( |
| **25%** | 1825.500000 | 20.000000 | 58.000000 | 7391.000000 | 5.000000 | 5.000000 | 1953.00( |
| **50%** | 2190.000000 | 50.000000 | 67.000000 | 9399.000000 | 6.000000 | 5.000000 | 1973.00( |
| **75%** | 2554.500000 | 70.000000 | 80.000000 | 11517.500000 | 7.000000 | 6.000000 | 2001.00( |
| **max** | 2919.000000 | 190.000000 | 200.000000 | 56600.000000 | 10.000000 | 9.000000 | 2010.00( |

Miêu tả dữ liệu cho thấy dữ liệu tất cả đều mang giá trị dương (>0) nên có thể chính xác về mặt giá trị. Nhưng vẫn còn nhiều trường dữ liệu có giữ liệu trống hoặc không liên quan cần loại bỏ.

```
df.isnull().sum()
```

```
Id               0 HalfBath          0 GarageQual       78
MSSubClass       0 BedroomAbvGr      0 GarageCond       78
MSZoning         4 KitchenAbvGr      0 PavedDrive        0
LotFrontage    227 KitchenQual       1 WoodDeckSF        0
LotArea          0 TotRmsAbvGrd      0 OpenPorchSF       0
Street           0 Functional        2 EnclosedPorch     0
Alley         1352 Fireplaces        0 3SsnPorch         0
LotShape         0 FireplaceQu     730 ScreenPorch       0
LandContour      0 GarageType       76 PoolArea          0
Utilities        2 GarageYrBlt      78 PoolQC         1456
LotConfig        0 GarageFinish     78 Fence          1169
LandSlope        0 GarageCars        1 MiscFeature    1408
Neighborhood     0 GarageArea        1 MiscVal           0
```

Vì là tệp dữ liệu về nhà ở nên ở mỗi trường dữ liệu đều có thêm các dữ liệu phân loại cụ thể, trong đó có rất nhiều dữ liệu trống và không cần thiết

```
df = df.dropna(axis=1)
print(df)
```

```
1442       Partial
1443       Partial
1444        Normal
1445        Normal
1446        Normal
1447        Normal
1448        Normal
1449        Normal
1450        Normal
1451        Normal
1452       Abnorml
1453        Normal
1454        Normal
1455       Abnorml
1456       Abnorml
1457        Normal
1458        Normal

[1459 rows x 47 columns]
```

Trước tiên, trong tệp dữ liệu có chứa các trường dữ liệu tồn tại giá trị null, chúng ta cần xác định dữ liệu đó có thật sự cần thiết và loại bỏ khỏi tệp dữ liệu. Sau khi xóa các cột chứa giá trị null, tệp dữ liệu còn lại 47 thuộc tính cùng 1459 bản ghi.

```
50    #Create a Random Forest
51    rf = tfdf.keras.RandomForestModel(task = tfdf.keras.Task.REGRESSION)
52    rf.compile(metrics=["mse"]) # Optional, you can use this to include a list of eval metrics
```

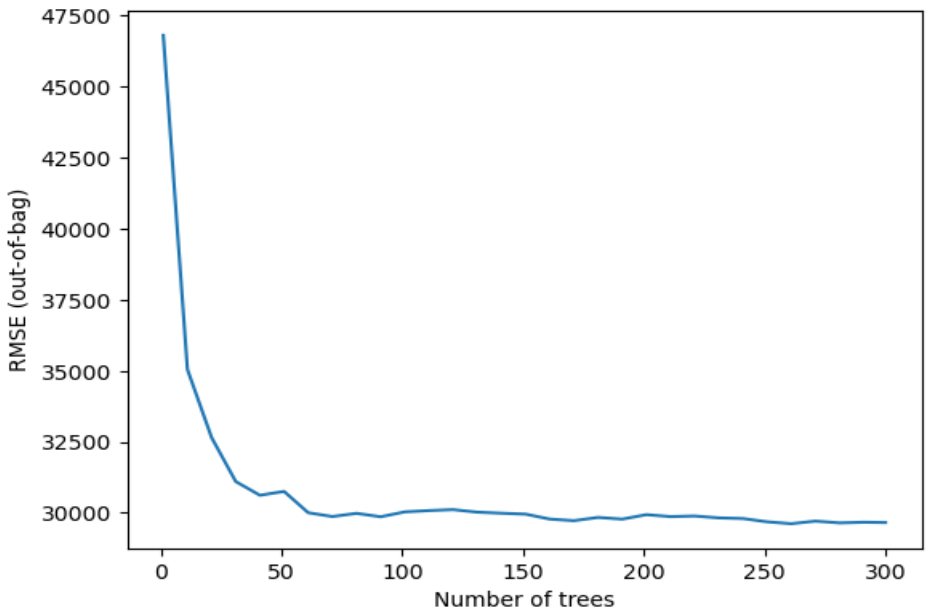
### 4.3. Đánh giá thang đo hiệu năng

Trong trường hợp này, sử dụng công việc REGRESSION cho biết mô hình bài toán sẽ được dùng để dự đoán giá trị liên tục (hồi quy). Sau khi tạo mô hình thành công, trong quá trình "compile" sẽ chỉ định tham số huấn luyện và các độ đo để đánh giá hiệu suất của mô hình. Độ đo phổ biến được sử dụng trong các bài toán hồi quy để đo lường sự khác biệt giữa giá trị dự đoán và giá trị thực tế là "mse" (Mean Squared Error)

```
54      #Train the model
55      rf.fit(x=train_ds)
```

Sau khi đã tạo mô hình, mô hình sẽ được huấn luyện bằng dữ liệu đầu vào "train_ds". Dữ liệu huấn luyện này sẽ được sử dụng để điều chỉnh các tham số của mô hình Random Forest sao cho mô hình có thể dự đoán đúng các đầu ra tương ứng với các đầu vào.

```
62      logs = rf.make_inspector().training_logs()
63      plt.plot([log.num_trees for log in logs], [log.evaluation.rmse for log in logs])
64      plt.xlabel("Number of trees")
65      plt.ylabel("RMSE (out-of-bag)")
66      plt.show()
67      inspector = rf.make_inspector()
68      inspector.evaluation()
69      evaluation = rf.evaluate(x=valid_ds,return_dict=True)
```



Biểu đồ ở trên cho thấy sự thay đổi các giá trị RMSE (Root Mean Square Error) khi them hoặc loại bỏ số lượng cây trong mô hình. Việc thêm nhiều cây có thể giúp tăng thêm hiệu suất nhưng cũng có thể gây ra overfitting trên tập dữ liệu huấn luyện

RMSE là một phép đo đánh giá mức độ chênh lệch giữa các giá trị dự đoán và các giá trị thực tế. Trong trường hợp này, RMSE được đánh giá từ nhật ký huấn luyện của mô hình Random Forest. Điểm số RMSE càng thấp, tức là mô hình dự đoán càng chính xác. Nếu có một điểm cụ thể của số lượng cây mà giá trị RMSE đạt đến mức thấp nhất hoặc ổn định, chúng ta có thể xác định được số lượng cây tốt nhất cho mô hình của mình

```
Out[18]:
        Evaluation(num_examples=1010, accuracy=None, loss=None, rmse=29660.363022492173, ndcg=None, auc
        s=None, auuc=None, qini=None)
```

Số lượng ví dụ trong tập dữ liệu thử nghiệm được sử dụng để đánh giá mô hình là 1010. Với chỉ số accuracy của mô hình là None, có nghĩa là mô hình không được đánh giá dựa trên độ chính xác. Tương tự như giá trị mất mát (loss) cũng là None.

Giá trị của Root Mean Square Error (RMSE) của mô hình trên tập dữ liệu thử nghiệm là RMSE=29660.363022492173 . Ngoài ra các chỉ số phụ đều đưa kết quả None có nghĩa là mô hình này không được đánh giá dựa trên các chỉ số đó. Giá trị RMSE = 29660.363022492173 cho thấy mô hình Decision Forests có độ chính xác tương đối thấp trong việc dự đoán giá nhà.

```
inspector.variable_importances()["NUM_AS_ROOT"]
```

```
[("OverallQual" (1; #62), 121.0),
 ("GarageCars" (1; #32), 49.0),
 ("ExterQual" (4; #22), 40.0),
 ("Neighborhood" (4; #59), 35.0),
 ("GrLivArea" (1; #38), 21.0),
 ("GarageArea" (1; #31), 15.0),
 ("BsmtQual" (4; #14), 7.0),
 ("YearBuilt" (1; #76), 5.0),
 ("KitchenQual" (4; #44), 4.0),
 ("TotalBsmtSF" (1; #73), 3.0)]
```

Đây là kết quả của biến có thứ tự quan trọng ví dụ như OverallQual là vị trí 1 trong danh sách các biến. Số 62 trong ngoặc đơn là chỉ số của biến này trong tập dữ liệu hoặc mô hình. Mức độ quan trọng của biến này được đo bằng giá trị 121.0.

OverallQual (Chất lượng tổng thể của căn nhà): Đây thường là một trong những biến quan trọng nhất khi dự đoán giá nhà. Mức độ quan trọng cao đề xuất rằng mô hình xem xét Chất lượng Tổng thể là yếu tố quan trọng trong việc dự đoán giá nhà. Mô hình có thể sử dụng biến này làm điểm bắt đầu để phân chia dữ liệu thành các nhóm nhỏ hơn dựa trên chất lượng tổng thể của căn nhà.

GarageCars (Số lượng xe trong ga-rơ): Đây cũng là một yếu tố quan trọng, vì số lượng xe có thể ảnh hưởng đến giá nhà. Một ga-rơ có thể được sử dụng để lưu trữ các phương tiện, và số lượng xe mà nó có thể chứa có thể ảnh hưởng đến giá trị của căn nhà.

ExterQual (Chất lượng bên ngoài): Chất lượng bên ngoài của căn nhà cũng có ảnh hưởng đáng kể đến giá trị của nó. Mô hình có thể sử dụng thông tin này để phân loại các căn nhà thành các nhóm có chất lượng bên ngoài khác nhau, từ đó giúp dự đoán giá trị của chúng.

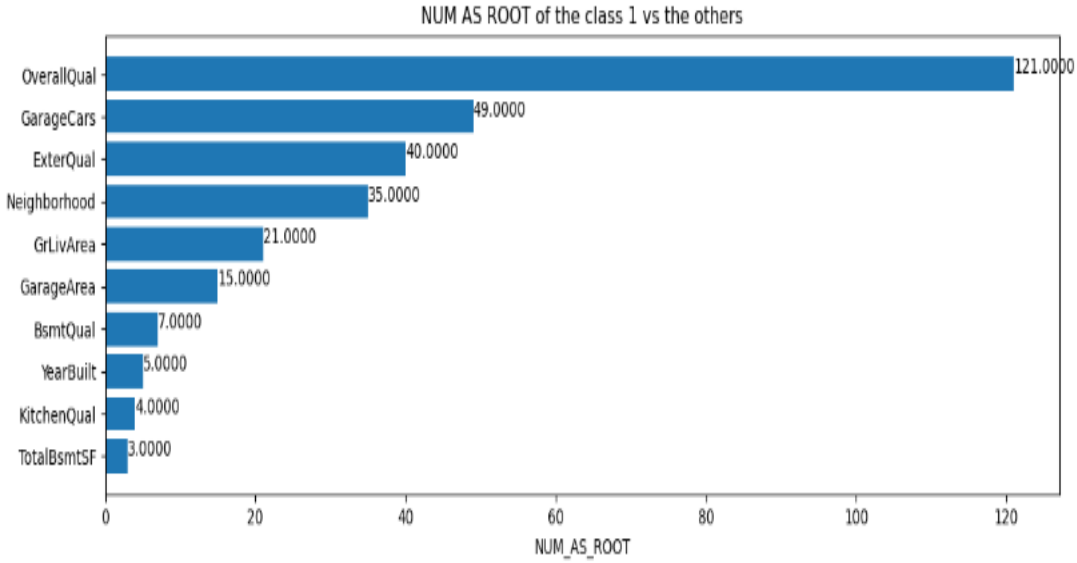
Neighborhood (Khu vực): Khu vực mà căn nhà nằm trong có thể ảnh hưởng lớn đến giá nhà. Mô hình có thể sử dụng thông tin này để xác định các khu vực có giá trị cao hơn và thấp hơn, từ đó dự đoán giá trị của các căn nhà trong khu vực đó.

Và những biến khác như GrLivArea, GarageArea, BsmtQual, YearBuilt, KitchenQual, và TotalBsmtSF cũng có ảnh hưởng đến giá nhà và được mô hình coi là quan trọng trong quá trình dự đoán.

```
for importance, patch in zip(feature_importances, bar.patches):
  plt.text(patch.get_x() + patch.get_width(), patch.get_y(), f"{importance:.4f}", va="top")

plt.xlabel(variable_importance_metric)
plt.title("NUM AS ROOT of the class 1 vs the others")
plt.tight_layout()
plt.show()
```

Chúng ta tạo một biểu đồ, trong đó trục tung là các biến quan trọng trong quá trình dự đoán còn trục hoành là thang đo độ quan trọng của chúng. Có thể thấy, chất lượng chung của căn nhà (OverallQual) là biến quan trọng nhất; gara ô tô, chất lượng ngoại thất hay hàng xóm là nhóm biến quan trọng tiếp theo; còn diện tích đất, diện tích gara, chất lượng móng căn nhà, thời gian xây dựng nhà, chất lượng căn bếp hay tổng diện tích dưới lòng đất không quá quan trọng.




* Tải dữ liệu kiểm tra:

```
test_file_path = "D:/QTDL with Spark/test.csv"
test_data = pd.read_csv(test_file_path)
ids = test_data.pop('Id')

test_ds = tfdf.keras.pd_dataframe_to_tf_dataset(
    test_data,
    task = tfdf.keras.Task.REGRESSION)

preds = rf.predict(test_ds)
output = pd.DataFrame({'Id': ids,
                       'SalePrice': preds.squeeze()})

output.head()
```

Mô hình RandomForest (rf) được sử dụng để dự đoán giá nhà trên dữ liệu kiểm tra. Một dataframe mới được tạo ra với 2 cột là ‘Id’ là các giá trị đã loại bỏ từ dữ liệu kiểm tra

và 'SalePrice' là các giá trị dự đoán được tính từ mô hình RandomForest. Sau khoảng hơn 1 giây, 5 dòng đầu của dataframe đó được thể hiện, cho thấy mức giá dự đoán của đối với từng căn nhà với những biến số khác nhau.

| | Id | SalePrice |
|---|---|---|
| 0 | 1461 | 123554.718750 |
| 1 | 1462 | 153939.062500 |
| 2 | 1463 | 176793.765625 |
| 3 | 1464 | 183828.296875 |
| 4 | 1465 | 193644.484375 |

### 4.4. Các tham số và môi trường cài đặt

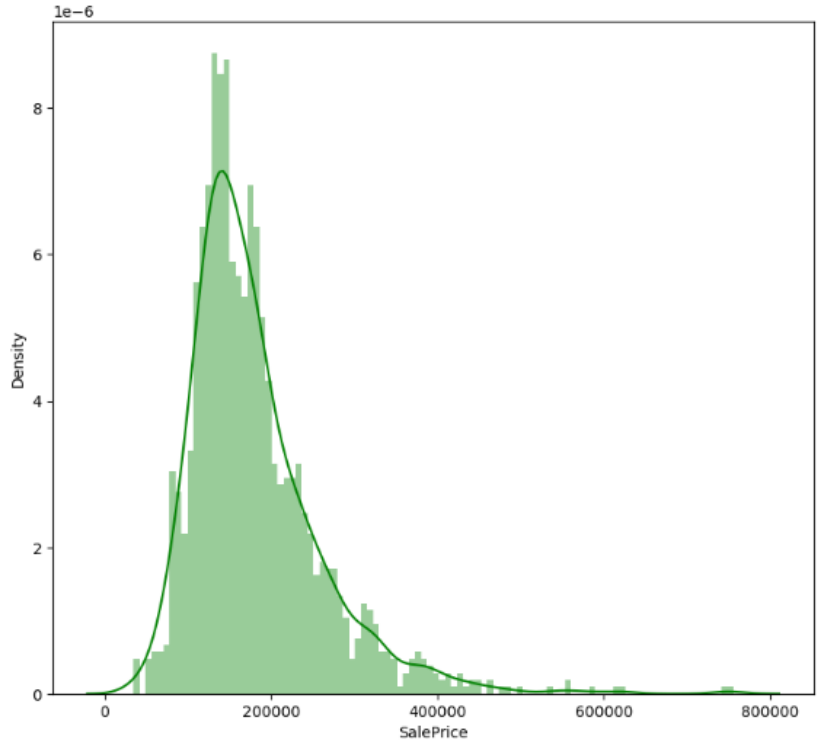
Môi trường cài đặt cho bài toán "House price prediction" thường bao gồm các công cụ và thư viện phổ biến được sử dụng cho xử lý dữ liệu và khai thác quy tắc kết hợp từ tập dữ liệu mua sắm. Dưới đây là các thành phần cần thiết cho bài toán này:

Ngôn ngữ lập trình: Nhóm sử dụng ngôn ngữ Python vì ngôn ngữ này cung cấp các thư viện mạnh mẽ cho xử lý dữ liệu và phân tích.

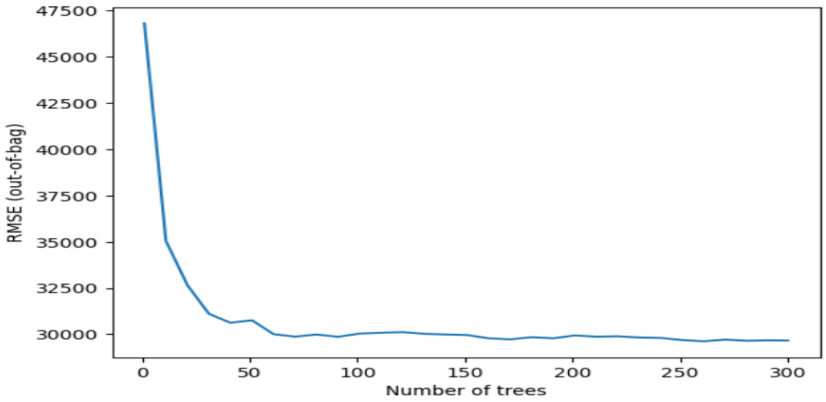
Nhóm tải các thư viện cần thiết cho việc phân tích dữ liệu và trực quan hóa dữ liệu bằng các biểu đồ như: pandas, seaborn, matplotlib, tensorflow

Thư viện Seaborn là một thư viện Python dùng để trực quan hóa dữ liệu dựa trên Matplotlib. Nó cung cấp một giao diện cấp cao để tạo ra các biểu đồ thống kê và trực quan hóa dữ liệu một cách dễ dàng và hiệu quả, bao gồm biểu đồ đường, biểu đồ cột, biểu đồ tròn,… Trong bài này, thư viện seaborn được cung cấp để vẽ biểu đồ displot.

```
In [7]:
print(dataset_df['SalePrice'].describe())
plt.figure(figsize=(9, 8))
sns.distplot(dataset_df['SalePrice'], color='g', bins=100, hist_kws={'alpha': 0.4});
```




Thư viện Matplotlib được cung cấp để tạo biểu đồ đường:




Các tham số được sử dụng trong bài toán “House price prediction”

Tập dữ liệu (Dataset): Đây là file dữ liệu gồm 80 trường dữ liệu khác nhau, mỗi trường thể hiện một thông số của căn nhà như: chất lượng căn nhà, diện tích nhà, phố, số lượng phòng,...

Biến quan trọng: Thể hiện các yếu tố khác nhau với mức độ quan trọng khác nhau, biến càng quan trọng thì mức độ tác động đến mô hình của nó càng lớn.

### 4.5. Phương pháp cơ sở

Đoạn code sử dụng quyết định rừng (Random Forest), một thuật toán học máy để dự đoán giá nhà. Random Forest  hoạt động dựa trên việc kết hợp nhiều cây quyết định (decision tree) đơn giản.

Mỗi cây quyết định được huấn luyện trên một tập hợp con (subset) của dữ liệu và đưa ra dự đoán độc lập. Dự đoán cuối cùng của mô hình là tổng hợp (aggregation) của các dự đoán từ các cây quyết định thành phần.

Random Forest có thể được coi là một phương pháp học tập tổng hợp (ensemble learning method) vì nó kết hợp nhiều mô hình yếu (individual weak models) để tạo ra một mô hình mạnh hơn.

Một số kỹ thuật để cải thiện hiệu quả của mô hình Random Forest:

- Sử dụng các kỹ thuật chọn lọc đặc trưng (feature selection) để chọn ra các đặc trưng quan trọng nhất cho mô hình.
- Sử dụng các kỹ thuật tăng cường dữ liệu (data augmentation) để tăng kích thước tập dữ liệu và giảm thiểu nguy cơ quá khớp (overfitting)
- Sử dụng các kỹ thuật ensemble để kết hợp nhiều mô hình Random Forest lại với nhau để tạo ra một mô hình mạnh mẽ hơn.

### 4.6. Phân tích, so sánh các kết quả

So sánh kết quả của hai mô hình Hồi quy tuyến tính và TensorFlow Decision Forests

Thời gian train:

- Hồi quy tuyến tính: Thường train nhanh hơn TensorFlow Decision Forests do đơn giản hơn.
- TensorFlow Decision Forests: Thường train chậm hơn do phức tạp hơn và cần xử lý nhiều dữ liệu hơn.

Độ chính xác:

- Hồi quy tuyến tính: Độ chính xác thấp hơn TensorFlow Decision Forests khi mối quan hệ giữa các biến là phi tuyến tính.
- TensorFlow Decision Forests: Có khả năng học các mối quan hệ phi tuyến tính phức tạp, do đó có thể đạt độ chính xác cao hơn.

## IV. Kết luận

Tùy vào kích thước tệp dữ liệu và các biến cần xử lý thì  bạn có thể lựa chọn giữa hai mô hình

Lựa chọn mô hình:

- Hồi quy tuyến tính: Phù hợp cho tập dữ liệu nhỏ , có thể sử dụng cho các biến có mối quan hệ tuyến tính với giá bán nhà.
- TensorFlow Decision Forests: Phù hợp cho tập dữ liệu lớn, có thể sử dụng cho các biến có mối quan hệ phi tuyến tính với giá bán nhà, cũng như để xử lý các tương tác phức tạp giữa các biến.

Nên sử dụng TensorFlow Decision Forests model dự đoán giá nhà với tập dữ liệu trên

Lý do:

- Có nhiều biến có mối quan hệ phi tuyến tính với giá bán nhà.
- TensorFlow Decision Forests có khả năng xử lý các tương tác phức tạp giữa các biến.