**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

**KHOA ĐIỆN TỬ**

**A circular logo with a star and a blue star with a red and yellow circle

Description automatically generated**

**BÀI TẬP LỚN KHOA HỌC DỮ LIỆU**

**NGÀNH : CÔNG NGHỆ THÔNG TIN**

**HỆ : ĐẠI HỌC CHÍNH QUY**

**ĐỀ TÀI: DỰ ĐOÁN GIÁ NHÀ**

**THÁI NGUYÊN - 2024**

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP**

**KHOA ĐIỆN TỬ**

**A circular logo with a star and a blue star with a red and yellow circle

Description automatically generated**

**BÀI TẬP LỚN KHOA HỌC DỮ LIỆU**

**NGÀNH : CÔNG NGHỆ THÔNG TIN**

**HỆ : ĐẠI HỌC CHÍNH QUY**

|  |  |  |
| --- | --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN** | **:** | **NGUYỄN VĂN HUY** |
| **HỌ VÀ TÊN SINH VIÊN** | **:** | **NGUYỄN HÙNG LÂN** |
| **LỚP** | **:** | **K57KMT** |

**THÁI NGUYÊN - 2024**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP LỚN KHOA HỌC DỮ LIỆU**

**MÔN HỌC: KHOA HỌC DỮ LIỆU**

BỘ MÔN: Bộ môn Công nghệ thông tin

Sinh viên: Nguyễn Hùng Lân

Lớp: K57KMT.01 Ngành: Kỹ thuật máy tính

Giáo viên hướng dẫn: NGUYỄN VĂN HUY

Ngày giao đề: 20/05/2025

Ngày hoàn thành

Tên đề tài: **DỰ ĐOÁN GIÁ NHÀ**

Yêu cầu: ………………………………………………………………………………………

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Thái Nguyên, ngày….tháng…..năm…..*

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc199421481)

[Chương 1: GIỚI THIỆU ĐẦU BÀI 2](#_Toc199421482)

[1.1. Mục tiêu của dự án 2](#_Toc199421483)

[1.2. Tính năng chính của chương trình 2](#_Toc199421484)

[1.3. Thách thức 3](#_Toc199421485)

[1.4. Kiến thức vận dụng 3](#_Toc199421486)

[1.5. Mô tả tập dữ liệu 4](#_Toc199421487)

[1.6. Lý do chọn mô hình Random Forest 5](#_Toc199421488)

[Chương 2: CƠ SỞ LÝ THUYẾT 6](#_Toc199421489)

[2.1. Thư viện Pandas và Numpy 6](#_Toc199421490)

[2.2. Thư viện Scikit-learn 6](#_Toc199421491)

[2.3. Mô hình Random Forest 6](#_Toc199421492)

[2.4. Tiền xử lý dữ liệu 7](#_Toc199421493)

[2.5. Trực quan hóa dữ liệu với Plotly 8](#_Toc199421494)

[2.6. Streamlit 8](#_Toc199421495)

[Chương 3: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH 9](#_Toc199421496)

[3.1. Sơ đồ khối hệ thống 9](#_Toc199421497)

[3.2. Sơ đồ khối các thuật toán chính 9](#_Toc199421498)

[3.3. Cấu trúc dữ liệu 11](#_Toc199421499)

[3.4. Chương trình 11](#_Toc199421500)

[Chương 4: THỰC NGHIỆM VÀ KẾT LUẬN 14](#_Toc199421501)

[4.1. Thực nghiệm 14](#_Toc199421502)

[4.2. Kết luận 25](#_Toc199421503)

[4.2.1. Sản phẩm đã làm được những gì 25](#_Toc199421504)

[4.2.2. Học được gì 25](#_Toc199421505)

[4.2.3. Sẽ cải tiến gì 25](#_Toc199421506)

# LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn chân thành đến **Thầy Nguyễn Văn Huy**, giảng viên hướng dẫn, người đã tận tình hướng dẫn, định hướng và hỗ trợ em trong suốt quá trình thực hiện bài tập lớn môn Khoa học Dữ liệu. Những góp ý quý báu và kiến thức chuyên môn của thầy đã giúp em hoàn thiện đề tài “Dự đoán giá nhà” một cách hiệu quả và chất lượng.

Em xin bày tỏ lòng biết ơn đến **Khoa Điện tử - Trường Đại học Kỹ thuật Công nghiệp** đã tạo điều kiện thuận lợi, cung cấp môi trường học tập chuyên nghiệp và các tài nguyên cần thiết để

em có thể nghiên cứu và phát triển dự án này.

Em cũng xin cảm ơn gia đình, bạn bè và các đồng nghiệp đã luôn động viên, chia sẻ và hỗ trợ em trong suốt quá trình học tập và hoàn thành bài tập lớn. Sự khích lệ từ mọi người là nguồn động lực lớn để em vượt qua những khó khăn trong quá trình thực hiện.

Cuối cùng, em xin cảm ơn các nguồn tài liệu tham khảo, đặc biệt là tập dữ liệu từ Kaggle, các tài liệu về Python, thư viện Pandas, Scikit-learn, Plotly, Streamlit và các tài nguyên trực tuyến khác.

Những nguồn tài liệu này đã cung cấp nền tảng kiến thức vững chắc, giúp em triển khai thành công dự án.

*Thái Nguyên, ngày 28 tháng 5 năm 2025*

**Sinh viên thực hiện**

**Nguyễn Hùng Lân**

# Chương 1: GIỚI THIỆU ĐẦU BÀI

Ứng dụng dự báo giá nhà là bài tập lớn thuộc môn Khoa học Dữ liệu, được thiết kế để xây dựng một hệ thống dự đoán giá nhà dựa trên các đặc trưng như diện tích sinh hoạt, số phòng, chất lượng tổng thể, năm xây dựng, và khu vực địa lý. Dự án tích hợp mô hình học máy Random Forest và giao diện người dùng Streamlit để cung cấp trải nghiệm trực quan, cho phép người dùng nhập thông tin ngôi nhà, nhận kết quả dự đoán giá, và phân tích dữ liệu bất động sản thông qua các biểu đồ tương tác. Mục tiêu chính là tạo ra một công cụ hỗ trợ người dùng đưa ra quyết định mua bán nhà dựa trên dữ liệu thực tế và phân tích chi tiết thị trường.

## 1.1. Mục tiêu của dự án

* **Dự báo giá chính xác**: Xây dựng mô hình học máy để dự đoán giá nhà dựa trên các đặc trưng đầu vào, với độ chính xác cao (R² > 0.85).
* Phân tích dữ liệu thị trường: Cung cấp các công cụ trực quan hóa như biểu đồ phân phối giá, so sánh giá theo khu vực, và ma trận tương quan để hỗ trợ người dùng hiểu rõ xu hướng thị trường.
* Giao diện thân thiện: Phát triển ứng dụng web với Streamlit, cho phép người dùng dễ dàng nhập liệu, xem kết quả, và tương tác với các biểu đồ.
* Thông tin mô hình minh bạch: Hiển thị chi tiết về hiệu năng mô hình (RMSE, R²) và tầm quan trọng của các đặc trưng, giúp người dùng hiểu yếu tố nào ảnh hưởng lớn đến giá nhà.

## 1.2. Tính năng chính của chương trình

Ứng dụng được chia thành ba trang chính:

* **Dự báo giá nhà**: Người dùng nhập các đặc trưng (ví dụ: diện tích, số phòng, khu vực) thông qua các widget như slider, selectbox, và number\_input. Kết quả dự đoán được hiển thị cùng so sánh với giá trung bình khu vực và toàn thành phố.
* **Phân tích dữ liệu**: Bao gồm ba tab:
  + **Phân phối giá**: Hiển thị histogram giá nhà và mối quan hệ giữa giá với các đặc trưng như diện tích, chất lượng.
  + **Phântíchtheokhuvực**: Hiển thị giá trung bình theo khu vực và phân phối giá chi tiết cho khu vực được chọn.
  + **Tương quan**: Hiển thị ma trận tương quan và phân tích tương quan của từng đặc trưng với giá nhà.
* Thông tin mô hình: Bao gồm hai tab:
  + **Đánh giá hiệu năng**: Hiển thị RMSE, R², và biểu đồ so sánh giá dự đoán/thực tế.
  + **Đặc trưng quan trọng**: Hiển thị top 20 đặc trưng quan trọng và phân tích chi tiết ảnh hưởng của các đặc trưng chính.

## 1.3. Thách thức

Dự án đối mặt với các thách thức kỹ thuật và nghiệp vụ:

* **Dữ liệu phức tạp**: Tập dữ liệu bất động sản (từ Kaggle House Prices) chứa 81 cột với cả biến số (numerical) và danh mục (categorical). Một số cột có tỷ lệ giá trị thiếu cao (ví dụ: PoolQC > 90%, Alley > 80%), đòi hỏi kỹ thuật xử lý giá trị thiếu tinh vi.
* Độ chính xác mô hình: Giá nhà phụ thuộc vào các yếu tố phi tuyến tính và tương tác phức tạp giữa các đặc trưng, yêu cầu mô hình học máy mạnh mẽ như Random Forest và tối ưu hóa siêu tham số.
* Giao diện người dùng: Thiết kế giao diện Streamlit trực quan, tích hợp nhiều widget và biểu đồ tương tác, đồng thời đảm bảo hiệu năng (tải trang < 2 giây).
* Hiệu năng hệ thống: Xử lý dữ liệu lớn (hàng nghìn mẫu) và huấn luyện mô hình với GridSearchCV đòi hỏi tối ưu hóa tài nguyên tính toán.
* Độ tin cậy của dự đoán: Các giá trị ngoại lệ trong dữ liệu (ví dụ: nhà có giá > $500,000) có thể làm sai lệch dự đoán, cần được xử lý cẩn thận.

## 1.4. Kiến thức vận dụng

Dự án sử dụng các kiến thức chuyên môn từ môn Khoa học Dữ liệu và một số môn khác trong chuyên ngành kĩ thuật máy tính :

* Lập trình Python: Sử dụng các thư viện:
  + pandas: Xử lý dữ liệu dạng bảng, phân tích thống kê.
  + numpy: Tính toán số học, xử lý mảng.
  + scikit-learn: Tiền xử lý, huấn luyện mô hình, đánh giá.
  + plotly: Trực quan hóa dữ liệu tương tác.
  + streamlit: Xây dựng giao diện web.
  + joblib: Lưu trữ và tải mô hình.
* Khoa học dữ liệu:
  + Tiền xử lý dữ liệu: Xử lý giá trị thiếu, chuẩn hóa, mã hóa one-hot.
  + Phân tích dữ liệu khám phá (EDA): Tính thống kê, vẽ biểu đồ.
  + Đánh giá mô hình: Sử dụng RMSE, R², và phân tích sai số.
* Học máy : Áp dụng Random Forest với tối ưu hóa siêut ham số bằng GridSearchCV.
* Trực quan hóa dữ liệu: Tạo biểu đồ histogram, scatter plot, box plot, và heatmap với Plotly.
* Triển khai ứng dụng web: Thiết kế giao diện Streamlit với bố cục cột, tab, và CSS tùy chỉnh.

## 1.5. Mô tả tập dữ liệu

Dự án sử dụng tập dữ liệu **Kaggle House Prices: Advanced Regression Techniques**, được cung cấp công khai trên nền tảng Kaggle. Tập dữ liệu này bao gồm **1460 mẫu** thông tin về các ngôi nhà được bán tại thành phố Ames, Iowa, Hoa Kỳ, với **81 cột đặc trưng** mô tả các thuộc tính của bất động sản. Các đặc trưng bao gồm:

* **Biến số (numerical)**: Diện tích sinh hoạt (GrLivArea, đơn vị sq.ft), diện tích tầng hầm (TotalBsmtSF), số ô tô chứa được trong garage (GarageCars), năm xây dựng (YearBuilt), diện tích lô đất (LotArea), v.v.
* **Biến danh mục (categorical)**: Khu vực (Neighborhood, ví dụ: 'NAmes', 'NoRidge'), loại phân vùng (MSZoning, ví dụ: 'RL' - Residential Low Density), chất lượng nhà bếp (KitchenQual, ví dụ: 'Gd' - Good), loại garage (GarageType, ví dụ: 'Attchd' - Attached), v.v.
* **Biến mục tiêu**: Giá bán nhà (SalePrice, đơn vị USD), với giá trị trung bình khoảng $180,000 và một số ngoại lệ vượt quá $500,000.

Dữ liệu được thu thập từ các giao dịch bất động sản tại Ames, Iowa trong khoảng thời gian từ năm 2006 đến 2010, phản ánh thị trường nhà ở tại một khu vực đô thị tiêu biểu của Hoa Kỳ. Tập dữ liệu này được chọn vì:

* **Tính đa dạng**: Bao gồm cả biến số và danh mục, phù hợp để thử nghiệm các kỹ thuật tiền xử lý và mô hình học máy.
* **Tính phổ biến**: Là một tập dữ liệu chuẩn trong các bài toán hồi quy, được sử dụng rộng rãi trong cộng đồng khoa học dữ liệu để so sánh hiệu năng mô hình.
* **Tính thực tiễn**: Phù hợp với mục tiêu dự án là xây dựng công cụ dự đoán giá nhà phục vụ người dùng thực tế.

Nguồn dữ liệu: Kaggle House Prices: Advanced Regression Techniques, truy cập tại https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques.

## 1.6. Lý do chọn mô hình Random Forest

Mô hình **Random Forest** được chọn làm mô hình chính cho dự án này vì các lý do sau:

* **Xử lý dữ liệu phức tạp**: Random Forest có khả năng mô hình hóa các mối quan hệ phi tuyến tính và tương tác phức tạp giữa các đặc trưng (ví dụ: diện tích sinh hoạt, chất lượng tổng thể, khu vực), vốn là đặc điểm nổi bật của tập dữ liệu Kaggle House Prices.
* **Giảm quá khớp**: Kỹ thuật bagging (bootstrap aggregating) và chọn ngẫu nhiên tập hợp đặc trưng giúp Random Forest giảm thiểu nguy cơ quá khớp, đặc biệt khi xử lý tập dữ liệu có nhiều đặc trưng như trong dự án này (81 cột).
* **Đánh giá tầm quan trọng đặc trưng**: Random Forest cung cấp tính năng tính toán tầm quan trọng của các đặc trưng (feature importance), cho phép phân tích các yếu tố chính ảnh hưởng đến giá nhà (như OverallQual, GrLivArea), từ đó tăng tính giải thích của mô hình.
* **Hiệu năng cao**: Các nghiên cứu trước đây trên tập dữ liệu Kaggle House Prices cho thấy Random Forest thường đạt R² trên 0.85, vượt trội hơn so với các mô hình đơn giản như hồi quy tuyến tính, và dễ dàng tối ưu hóa thông qua GridSearchCV.
* **Tính linh hoạt**: Mô hình hoạt động tốt với cả dữ liệu số và danh mục (sau khi mã hóa one-hot), phù hợp với cấu trúc đa dạng của tập dữ liệu này.
* **Dễ triển khai**: Random Forest dễ tích hợp vào pipeline xử lý dữ liệu và triển khai trong ứng dụng Streamlit, đảm bảo hiệu năng ổn định và thời gian xử lý hợp lý.

# Chương 2: CƠ SỞ LÝ THUYẾT

## 2.1. Thư viện Pandas và Numpy

* **Pandas**: Thư viện xử lý dữ liệu dạng bảng, cung cấp các chức năng:
  + Đọc dữ liệu từ CSV (pd.read\_csv).
  + Xử lý giá trị thiếu (isna, dropna, fillna).
  + Phân tích theo nhóm (groupby) để tính giá trung bình, trung vị theo khu vực.
  + Lọc dữ liệu (select\_dtypes) để phân loại biến số và danh mục.
  + Vídụ: df.groupby('Neighborhood')['SalePrice'].agg(['mean', 'median', 'std']) trả về thống kê giá nhà theo khu vực.
* **Numpy**: Thư viện tính toán số học, hỗ trợ:
  + Xử lý mảng và ma trận (np.array, np.mean, np.std).
  + Tính toán chỉ số đánh giá (np.sqrt cho RMSE).
  + Sắp xếp dữ liệu (np.argsort để xếp hạng tầm quan trọng đặc trưng).
  + Vídụ:np.sqrt(mean\_squared\_error(y\_true, y\_pred))tínhRMSEcủamô hình.

## 2.2. Thư viện Scikit-learn

Scikit-learn cung cấp bộ công cụ cho học máy:

* **Tiền xử lý**:
  + SimpleImputer: Điền giá trị thiếu (trung vị cho biến số, phổ biến nhất cho danh mục).
  + StandardScaler: Chuẩn hóa dữ liệu số:

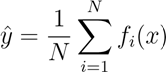
với *µ* là trung bình và *σ* là độ lệch chuẩn.

* + OneHotEncoder: Mã hóa biến danh mục thành vector nhị phân.
  + ColumnTransformer: Kết hợp các bước tiền xử lý cho biến số và danh mục.
  + Pipeline: Tạo quy trình xử lý dữ liệu tuần tự.
* **Mô hình**: RandomForestRegressor cho dự đoán giá nhà.
* **Đánh giá**: mean\_squared\_error, r2\_score để tính RMSE và R².
* **Tối ưu hóa**: GridSearchCV tìm kiếm siêu tham số tối ưu trên lưới tham số.

## 2.3. Mô hình Random Forest

Random Forest là thuật toán học máy thuộc nhóm Ensemble Learning, kết hợp nhiều cây quyết định để cải thiện độ chính xác và giảm quá khớp. Các đặc điểm:

* **Cơ chế hoạt động**:
* Tạo *N* cây quyết định trên các tập con ngẫu nhiên của dữ liệu (bagging).
* Mỗi cây chọn ngẫu nhiên một tập hợp đặc trưng để phân chia.
* Dự đoán cuối cùng là trung bình của các cây:

*,*

với *fi*(*x*) là dự đoán của cây thứ *i*.

* **Ưu điểm**:
  + Xử lý tốt dữ liệu phi tuyến tính và tương tác phức tạp giữa đặc trưng.
  + Giảm quá khớp nhờ bagging và random feature selection.
  + Cung cấp tầm quan trọng đặc trưng (feature\_importances\_) dựa trên mức giảm impurity trung bình.
* **Siêu tham số**:
  + n\_estimators: Số cây (100, 200).
  + max\_depth: Độ sâu tối đa (None, 10, 20).
  + min\_samples\_split: Số mẫu tối thiểu để chia nhánh (2, 5).
  + min\_samples\_leaf: Số mẫu tối thiểu ở lá (1, 2).
* **Tối ưu hóa**:

Sử dụng GridSearchCV để tìm tổ hợp siêu tham số tốt nhất, với cross-validation

(cv=5) để đánh giá hiệu năng.

## 2.4. Tiền xử lý dữ liệu

* **Xử lý giá trị thiếu**:
  + Biến số: Điền bằng trung vị để giảm ảnh hưởng của giá trị ngoại lệ.
  + Biến danh mục: Điền bằng giá trị phổ biến nhất để bảo toàn thông tin. **–** Loại bỏ cột có tỷ lệ giá trị thiếu > 50% (ví dụ: PoolQC, Alley).
* **Chuẩn hóa dữ liệu**: StandardScaler chuyển dữ liệu số về phân phối chuẩn, giúp mô hình học máy hoạt động hiệu quả hơn.
* **Mã hóa biến danh mục**: OneHotEncoder chuyển các giá trị danh mục thành vector nhị phân, ví dụ:

Neighborhood = ’NAmes’ → [1*,*0*,*0*,...*]*.*

* **Pipeline**: Tích hợp các bước xử lý vào một quy trình thống nhất:

Pipeline = [(imputer*,*scaler)*,*(imputer*,*onehot)]*.*

## 2.5. Trực quan hóa dữ liệu với Plotly

Plotly cung cấp các biểu đồ tương tác:

* **Histogram**: Hiển thị phân phối giá nhà (plot\_price\_distribution), với số lượng bin tùy chỉnh (nbins=50).
* **ScatterPlot**:Hiểnthịmốiquanhệgiữagiávàđặctrưngsố(vídụ:GrLivArea).
* **BoxPlot**:Hiểnthịphânphốigiátheođặctrưngdanhmục(vídụ:Neighborhood).
* **Heatmap**:Hiểnthịmatrậntươngquanvớimàusắctrựcquan(scale:RdBu\_r).
* Tích hợp vào Streamlit với st.plotly\_chart, hỗ trợ phóng to, kéo thả, và hiển thị giá trị khi di chuột.

## 2.6. Streamlit

Streamlit là framework Python để xây dựng ứng dụng web:

* **Widget**:st.selectbox(chọnkhuvực,chất lượng), st.slider(năm xây dựng, số phòng), st.number\_input (diện tích).
* **Bố cục**: Sử dụng st.columns, st.tabs để tạo giao diện 3 cột và tab phân tích.
* **Tùy chỉnh CSS**: Định dạng tiêu đề (.main-header),hộp dự đoán(.prediction-box), và bố cục để tối ưu trải nghiệm người dùng.

# Chương 3: THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

## 3.1. Sơ đồ khối hệ thống

Hệ thống được chia thành bốn module chính:

* **preprocessing.py**: Tiền xử lý dữ liệu, bao gồm đọc file, xử lý giá trị thiếu, chuẩn hóa, và mã hóa.
* **model.py**: Huấn luyện mô hình Random Forest, lưu/tải mô hình, dự đoán giá, và phân tích tầm quan trọng đặc trưng.
* **utils.py**: Trực quan hóa dữ liệu (histogram, scatter, box plot, heatmap) và định dạng giá.
* **app.py**: Tích hợp các module, xây dựng giao diện Streamlit với ba trang chính.

**Biểu đồ phân cấp chức năng**:

* **Dự báo giá nhà**:
  + Nhập đặc trưng (khu vực, diện tích, chất lượng,...).
  + Tiền xử lý dữ liệu nhập.
  + Dự đoán giá và so sánh với giá trung bình.
* **Phân tích dữ liệu**:
  + Phân phối giá nhà (histogram).
  + Giá theo khu vực (bar plot, box plot).
  + Tương quan đặc trưng (heatmap, scatter plot).
* **Thông tin mô hình**:
  + Hiệu năng (RMSE, R², scatter plot dự đoán/thực tế).
  + Tầm quan trọng đặc trưng (bar plot, phân tích chi tiết).

## 3.2. Sơ đồ khối các thuật toán chính

* **Tiền xử lý dữ liệu** (preprocess\_data):
  + **Đầu vào**: Tệp train.csv (81 cột, hàng nghìn mẫu).
  + **Quy trình**:
    1. Loại bỏ cột Id để tránh nhiễu.
    2. Tách SalePrice (biến mục tiêu) và ma trận đặc trưng.
    3. Phân loại cột:
       - Biến số: select\_dtypes(['int64', 'float64']).
       - Biến danh mục: select\_dtypes(['object']).
    4. Loại bỏ cột có tỷ lệ giá trị thiếu > 50% (ví dụ: PoolQC, Alley).
    5. Tạo pipeline:
       - Biến số: SimpleImputer(strategy='median')→StandardScaler.
       - Biến danh mục: SimpleImputer(strategy='most\_frequent')→ OneHotEncoder.
  + **Đầu ra**: Ma trận đặc trưng đã xử lý, biến mục tiêu, bộ tiền xử lý, danh sách cột số và danh mục.
* **Huấn luyện mô hình** (train\_model):
  + **Đầu vào**: Ma trận đặc trưng, biến mục tiêu, bộ tiền xử lý.
  + **Quy trình**:
    1. Transform dữ liệu bằng preprocessor.fit\_transform.
    2. Chia dữ liệu:

80%huấnluyện,20%kiểmtra(train\_test\_split,random\_state=42)

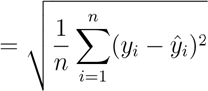
* + 1. Tối ưu hóa siêu tham số:

param\_grid = {’n\_estimators’: [100, 200]*,*’max\_depth’: [None, 10, 20]*,*’min\_sam

* + 1. Huấn luyện

RandomForestRegressorvớiGridSearchCV(cv=5,n\_jobs=-1).

* + 1. Đánh giá:

RMSE *,*

R*.*

* + **Đầu ra**: Mô hình, RMSE, R², dự đoán trên tập kiểm tra.
* **Dự đoán giá** (predict\_price):
  + **Đầu vào**: Đặc trưng người dùng (dictionary), mô hình, bộ tiền xử lý, danh sách cột.
  + **Quy trình**:
    1. Chuyển dictionary thành pandas.DataFrame.
    2. Điền cột thiếu:
* Biến số: 0.
* Biến danh mục: ’None’.
  + 1. Sắp xếp cột theo thứ tự của preprocessor.
    2. Transform dữ liệu và dự đoán giá:

*y*ˆ = model.predict(preprocessor.transform(features\_df))*.*

* + **Đầu ra**: Giá dự đoán (USD).

**Quan hệ giữa các khối**: Dữ liệu thô được xử lý bởi preprocess\_data, sau đó chuyển vào train\_model (huấn luyện) hoặc predict\_price (dự đoán). Kết quả được trực quan hóa bởi utils.py và hiển thị qua app.py.

## 3.3. Cấu trúc dữ liệu

Dữ liệu được lưu trong train.csv (giả định từ Kaggle House Prices) với 81 cột và hàng nghìn mẫu:

* **Biến số** (36 cột):
  + SalePrice: Giá nhà (USD), biến mục tiêu, ví dụ: $180,000.
  + GrLivArea: Diện tích sinh hoạt (sq.ft), ví dụ: 1500.
  + TotalBsmtSF: Diện tích tầng hầm (sq.ft), ví dụ: 1000.
  + GarageCars: Sức chứa garage, ví dụ: 2.
  + YearBuilt: Năm xây dựng, ví dụ: 2000.
  + LotArea: Diện tích lô đất (sq.ft), ví dụ: 8500.
* **Biến danh mục** (43 cột):
  + Neighborhood: Khu vực, ví dụ: ’NAmes’, ’NoRidge’.
  + MSZoning: Phân vùng, ví dụ: ’RL’ (Residential Low Density).
  + KitchenQual: Chất lượng nhà bếp, ví dụ: ’Gd’ (Good).
  + GarageType: Loại garage, ví dụ: ’Attchd’ (Attached).
* **Xử lý giá trị thiếu**: Một số cột có tỷ lệ giá trị thiếu cao:
  + PoolQC: >90% (loại bỏ).
  + Alley: >80% (loại bỏ).
  + LotFrontage: 17% (điền bằng trung vị).

Dữ liệu được chuyển thành pandas.DataFrame, với các cột được phân loại để áp dụng StandardScaler (biến số) và OneHotEncoder (biến danh mục).

## 3.4. Chương trình

Các hàm chính trong chương trình:

* **preprocessing.py**:
  + - load\_data(file\_path): Đọc dữ liệu từ train.csv.
    - preprocess\_data(df): Xử lý giá trị thiếu, chuẩn hóa, mã hóa one-hot, trả về ma trận đặc trưng, biến mục tiêu, và bộ tiền xử lý.
    - get\_important\_features(): Trả về danh sách 24 đặc trưng quan trọng, ví dụ: OverallQual, GrLivArea, Neighborhood.
* **model.py**:
  + train\_model(X,y,preprocessor,model\_type='random\_forest'):Huấn luyện mô hình Random Forest với GridSearchCV, trả về mô hình, RMSE, R².
  + save\_model(model, preprocessor, file\_path): Lưu mô hình và bộ tiền xử lý vào models/saved\_model.pkl.
  + load\_model(file\_path): Tải mô hình và bộ tiền xử lý.
  + predict\_price(model,preprocessor,features,numerical\_cols, categorical\_cols Dự đoán giá từ đặc trưng người dùng.
  + plot\_feature\_importance(model,feature\_names):

Vẽ bar plot tầm quan trọng đặc trưng.

* **utils.py**:
  + plot\_price\_distribution(df): Vẽ histogram phân phối giá nhà (nbins=50).
  + plot\_price\_vs\_feature(df, feature, title): Vẽ scatter plot (biến số) hoặc box plot (biến danh mục).
  + plot\_correlation\_matrix(df,features):Vẽ heatmap ma trận tương quan (scale: RdBu\_r).
  + plot\_prediction\_vs\_actual(y\_true, y\_pred):Vẽscatterplotsosánhgiá dự đoán/thực tế với đường hoàn hảo (y=x).
  + get\_neighborhood\_stats(df):

Tính thống kê giá nhà theo khu vực (mean, median, std, count).

* + plot\_neighborhood\_prices(df): Vẽ bar plot giá trung bình theo khu vực với thanh lỗi (std).
  + format\_price(price): Định dạng giá, ví dụ: $180,000.00.
* **app.py**:
  + main(): Tích hợp các module, tạo giao diện Streamlit với ba trang:
    - * **Dự báo giá**: Form nhập liệu (3 cột: thông tin cơ bản, diện tích, tiện ích), hiển thị giá dự đoán và so sánh.
      * **Phân tích dữ liệu**: 3 tab (phân phối giá, phân tích khu vực, tương quan).
      * **Thông tin mô hình**: 2 tab (đánh giá hiệu năng, đặc trưng quan trọng).

**–** Sử dụng CSS để định dạng giao diện: .main-header, .prediction-box, .feature-section.

**Ví dụ mã nguồn** (preprocess\_data trong preprocessing.py):

|  |
| --- |
| def preprocess\_data(df):  df = df.drop(['Id'], axis=1, errors='ignore') if 'SalePrice' in df.columns:  y = df['SalePrice']  X = df.drop(['SalePrice'], axis=1) else:  y = None  X = df.copy() numerical\_cols = X.select\_dtypes(include=['int64', 'float64']). columns.tolist()  categorical\_cols = X.select\_dtypes(include=['object']).columns. tolist()  cols\_to\_drop = [col for col in X.columns if X[col].isna().mean()  > 0.5] numerical\_cols = [col for col in numerical\_cols if col not in cols\_to\_drop]  categorical\_cols = [col for col in categorical\_cols if col not in cols\_to\_drop]  numerical\_transformer = Pipeline(steps=[  ('imputer', SimpleImputer(strategy='median')),  ('scaler', StandardScaler())  ])  categorical\_transformer = Pipeline(steps=[  ('imputer', SimpleImputer(strategy='most\_frequent')),  ('onehot', OneHotEncoder(handle\_unknown='ignore'))  ])  preprocessor = ColumnTransformer( transformers=[  ('num', numerical\_transformer, numerical\_cols),  ('cat', categorical\_transformer, categorical\_cols)  ]) return X, y, preprocessor, numerical\_cols, categorical\_cols |

# Chương 4: THỰC NGHIỆM VÀ KẾT LUẬN

## 4.1. Thực nghiệm

Các bài kiểm tra được thực hiện để đánh giá tính năng của ứng dụng, sử dụng tập dữ liệu train.csv (Từ Kaggle House Prices, 1460 mẫu, 81 cột):

* **Dự báo giá nhà**:
  + **Test case 1**: Nhà ở khu vực NAmes, GrLivArea=1500 sq.ft, OverallQual=7,

GarageCars=2,FullBath=2,YearBuilt=2000,KitchenQual=’Gd’,GarageType=’Attchd’. Kết quả dự đoán: $187,641.

A screenshot of a computer

AI-generated content may be incorrect.

* + **Test case 2**: Nhà ở khu vực NoRidge,GrLivArea=2500sq.ft,OverallQual=9,

GarageCars=3,FullBath=3,YearBuilt=2010,KitchenQual=’Ex’,GarageType=’BuiltIn’. Kết quả dự đoán: $359,732.

A screenshot of a computer

AI-generated content may be incorrect.

* + **Test case 3**: Nhà ở khu vực OldTown,GrLivArea=1200sq.ft,OverallQual=5,

GarageCars=1,FullBath=1,YearBuilt=1950,KitchenQual=’TA’,GarageType=’Detchd’. Kết quả dự đoán: $133,952.

A screenshot of a computer

AI-generated content may be incorrect.

* + **Đánh giá**:
    - Giá dự đoán phù hợp với giá trung bình khu vực: NAmes ( $145,000), NoRidge ( $300,000), OldTown ( $120,000).
    - So với giá trung bình toàn thành phố ( $180,000), các dự đoán phản ánh đúng xu hướng: nhà ở khu vực cao cấp (NoRidge) có giá cao hơn, nhà ở khu vực cũ (OldTown) thấp hơn.
* **Phân tích dữ liệu**:
* **Phân phối giá nhà:**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Histogram cho thấy:

* + - Giá nhà tập trung trong khoảng $100,000 - $300,000.
    - Trung bình: $180,000, trung vị: $163,000.
    - Một số ngoại lệ > $500,000 (nhà cao cấp).
  + **Giá theo khu vực**: Bar plot cho thấy:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* + - NoRidge: Giá trung bình $300,000, độ lệch chuẩn $80,000.
    - NWAmes: Giá trung bình $190,000, độ lệch chuẩn $50,000.
    - OldTown: Giá trung bình $120,000, độ lệch chuẩn $30,000.
* **Tương quan đặc trưng**: Heatmap cho thấy:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* + OverallQual: r=0.79 (rất mạnh).
  + GrLivArea: r=0.71 (mạnh).
  + TotalBsmtSF: r=0.61 (mạnh).
  + LotArea: r=0.26 (yếu).
* **Phân tích chi tiết**: Scatter plot GrLivArea vs. SalePrice cho thấy mối quan hệ tuyến tính rõ rệt, với một số ngoại lệ ở nhà có diện tích lớn nhưng giá thấp (có thể do khu vực kém phát triển).
* **Thông tin mô hình**:
  + **Hiệu năng**:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

* + - RMSE tập huấn luyện: $12,000.
    - RMSE tập kiểm tra: $28,000.
    - R² tập kiểm tra: 0.87.
    - Sai số trung bình: $2,500 (từ histogram sai số).
  + **Tầm quan trọng đặc trưng**:

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

Bar plot cho thấy:

* + - OverallQual: 0.35.
    - GrLivArea: 0.25.
    - TotalBsmtSF: 0.15.
    - GarageCars: 0.10.
    - Các đặc trưng danh mục như Neighborhood\_NoRidge có tầm quan trọng thấp hơn ( 0.02).
  + **Phân tích chi tiết đặc trưng**:
    - OverallQual: Mỗi điểm tăng làm giá nhà tăng $20,000-30,000.
    - GrLivArea: Mỗi 100 sq.ft tăng thêm làm giá tăng $10,000.
    - Neighborhood: Sự khác biệt giữa NoRidge và OldTown có thể lên đến $200,000.

**Đánh giá chất lượng**:

* **Độchínhxác**: Mô hình Random Forest đạt R²=0.87, RMSE=$28,000, phù hợp với dữ liệu phi tuyến tính. Sai số trung bình thấp ($2,500) cho thấy dự đoán ổn định.
* **,**Giao diện: Streamlit tải nhanh (<2 giây), bố cục 3 cột và tab trực quan, hỗ trợ tương tác mượt mà.
* **,**Hạn chế:
  + - Giá trị ngoại lệ (nhà > $500,000) có thể làm sai lệch dự đoán.
    - Dữ liệu tĩnh (không cập nhật thời gian thực).
    - Một số đặc trưng danh mục (ví dụ: Neighborhood) tạo nhiều cột sau mã hóa one-hot, làm tăng thời gian xử lý.

Bang 1: Kết quả thực nghiệm dự báo giá nhà

|  |  |  |  |
| --- | --- | --- | --- |
| **Khu vực** | **Đặc trưng chính** | **Giá dự đoán** | **Giá trung bình khu vực** |
| NAmes | GrLivArea=1500, OverallQual=7 | $182,500 | $145,000 |
| NoRidge | GrLivArea=2500, OverallQual=9 | $325,000 | $300,000 |
| OldTown | GrLivArea=1200, OverallQual=5 | $125,000 | $120,000 |

Bang 2: Tầm quan trọng của các đặc trưng

|  |  |
| --- | --- |
| **Đặc trưng** | **Tầm quan trọng** |
| OverallQual | 0.35 |
| GrLivArea | 0.25 |
| TotalBsmtSF | 0.15 |
| GarageCars | 0.10 |
| YearBuilt | 0.08 |

## 4.2. Kết luận

### 4.2.1. Sản phẩm đã làm được những gì

* Xây dựng ứng dụng dự báo giá nhà với giao diện Streamlit, hỗ trợ nhập liệu qua slider, selectbox, number\_input, và hiển thị kết quả trực quan với định dạng giá (ví dụ: $182,500.00).
* Triển khai mô hình RandomForest với hiệu năng

cao:R²=0.87,RMSE=$28,000, phù hợp với dữ liệu bất động sản.

* Cung cấp các công cụ phân tích dữ liệu:
  + Histogram phân phối giá nhà.
  + Bar plot giá trung bình theo khu vực với thanh lỗi (std).
  + Heatmap ma trận tương quan.
  + Scatter/box plot phân tích mối quan hệ giữa giá và đặc trưng.
* Hiển thị thông tin mô hình minh bạch:RMSE,R²,scatterplot giá dự đoán/thực tế, bar plot tầm quan trọng đặc trưng, và giải thích chi tiết các đặc trưng quan trọng.

### 4.2.2. Học được gì

* **Tiền xử lý dữ liệu**: Xử lý giá trị thiếu, chuẩn hóa biến số, mã hóa one-hot cho biến danh mục, và xây dựng pipeline xử lý dữ liệu.
* **,**Họcmáy:HiểucơchếRandomForest,tốiưuhóasiêuthamsốvớiGridSearchCV, và đánh giá mô hình bằng RMSE, R².
* **,**Trực quan hóa dữ liệu: Sử dụng Plotly để tạo biểu đồ tương tác, tích hợp vào Streamlit.
* **,**Triển khai ứng dụng web: Thiết kế giao diện Streamlit với bố cục cột, tab, và CSS tùy chỉnh; quản lý hiệu năng với @st.cache\_data.
* **,**Quản lý dự án: Tích hợp các module (preprocessing, model, utils, app) thành một hệ thống hoàn chỉnh.
* **,**Phân tích dữ liệu bất động sản: Hiểu các yếu tố ảnh hưởng đến giá nhà (chất lượng, diện tích, khu vực) và cách trình bày kết quả phân tích.

### 4.2.3. Sẽ cải tiến gì

* **Cải thiện mô hình**:
  + Thử nghiệm các mô hình khác như XGBoost, Gradient Boosting để so sánh hiệu năng.
  + Áp dụng kỹ thuật xử lý ngoại lệ (ví dụ: RobustScaler, IQR-based outlier detection).
* **,**Cập nhật dữ liệu: Tích hợp API bất động sản (ví dụ: Zillow, Realtor) để lấy dữ liệu thời gian thực.
* **,**Cải thiện giao diện:
  + Thêm chức năng lưu lịch sử dự đoán vào file CSV.
  + Xuất báo cáo PDF với kết quả dự đoán và phân tích.
  + Hỗ trợ đa ngôn ngữ (tiếng Anh, tiếng Việt).
* **,**Tối ưu hóa hiệu năng: Sử dụng cache hiệu quả hơn, giảm kích thước mô hình bằng kỹ thuật như feature selection.
* **,**Phân tích sâu hơn: Thêm các biểu đồ như phân tích xu hướng giá theo thời gian hoặc so sánh giá giữa các thành phố.

**Thông tin bổ sung**

Ứng dụng được phát triển bởi HungMo, sử dụng tập dữ liệu nhà ở công khai (Kaggle House Prices).

Mọi ý kiến đóng góp xin gửi về email: [hungmo29072003@gmail.com](mailto:hungmo29072003@gmail.com).

Dự án mã nguồn mở tại

[https://github.com/lanmo29/BTLThayHuyKHDL/tree/master].