BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM

BÁO CÁO HỌC PHẦN

TÊN HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI: XÂY DỰNG MÔ HÌNH MẠNG NEURON ĐỂ DỰ ĐOÁN GIÁ NHÀ.

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Lớp
1	1771020065	Nguyễn Thị Ngọc Ánh	05/12/2005	CNTT 17-04
2	1771020039	Vũ Tuyết Anh	05/10/2005	CNTT 17-04

Hà Nội, năm 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM

BÁO CÁO HỌC PHẦN

TÊN HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI: XÂY DỰNG MÔ HÌNH MẠNG NEURON ĐỂ DỰ ĐOÁN GIÁ NHÀ.

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bằng Số	Bằng Chữ
1	1771020065	Nguyễn Thị Ngọc Ánh	05/12/2005		
2	1771020039	Vũ Tuyết Anh	05/10/2005		

CÁN BỘ CHẤM THI 1 **CÁN BỘ CHẤM THI 2**

Hà Nội, năm 2025

LỜI NÓI ĐẦU

Trong bối cảnh thị trường bất động sản ngày càng phát triển, việc dự đoán giá nhà trở thành một bài toán quan trọng giúp người mua, người bán và các nhà đầu tư đưa ra quyết định hợp lý. Truyền thống, việc định giá nhà thường dựa trên kinh nghiệm của chuyên gia hoặc các phương pháp thống kê đơn giản, tuy nhiên, những cách tiếp cận này có thể không tận dụng được toàn bộ dữ liệu có sẵn và dẫn đến sai lệch trong dự báo.

Với sự phát triển mạnh mẽ của trí tuệ nhân tạo, mạng neuron nhân tạo (Artificial Neural Network – ANN) nổi lên như một công cụ hiệu quả trong việc mô hình hóa các mối quan hệ phức tạp giữa các yếu tố ảnh hưởng đến giá nhà. Đề tài này sử dụng mạng neuron để phân tích dữ liệu nhà đất dựa trên các đặc điểm như năm bán nhà, tuổi của ngôi nhà, khoảng cách đến trung tâm thành phố, số lượng cửa hàng trong khu vực và vị trí địa lý. Mô hình được huấn luyện trên dữ liệu thực tế, sau đó được đánh giá bằng các chỉ số lỗi nhằm kiểm tra độ chính xác và khả năng ứng dụng vào thực tế.

MỤC LỤC

MỤC LỤC HÌNH ẢNH

MỤC LỤC BẢNG

BẢNG CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	ANN	Artificial Neural Network

CHƯƠNG 1: GIỚI THIỆU VÀ PHÂN TÍCH YÊU CẦU BÀI TOÁN

1.1. Giới thiệu bài toán

1.1.1. Mô tả bài toán dự đoán giá nhà

Dự đoán giá nhà là một bài toán quan trọng trong lĩnh vực bất động sản, giúp người mua, người bán và các nhà đầu tư đưa ra quyết định hợp lý dựa trên dữ liệu lịch sử. Trong bài toán này, chúng ta sử dụng một tập dữ liệu chứa thông tin về nhiều căn nhà, bao gồm các đặc trưng như:

- Năm bán nhà.
- Tuổi của ngôi nhà tại thời điểm bán.
- Khoảng cách từ nhà đến trung tâm thành phố.
- Số lượng cửa hàng xung quanh.
- Vị trí địa lý của ngôi nhà (kinh độ, vĩ độ).
- Giá bán thực tế của nhà.

Mục tiêu chính của mô hình là xây dựng một mạng neuron nhân tạo (Artificial Neural Network - ANN) để dự đoán giá nhà dựa trên các đặc trưng trên.

1.1.2. Ứng dụng thực tế của mô hình trong định giá bất động sản

Mô hình dự đoán giá nhà có thể được ứng dụng trong nhiều lĩnh vực:

- Hỗ trợ người mua và người bán: Cung cấp giá nhà ước tính, giúp người mua tránh trả giá quá cao hoặc quá thấp.
- Hỗ trợ ngân hàng và tổ chức tài chính: Định giá tài sản thế chấp chính xác hơn khi cấp khoản vay.
- Hỗ trợ chính quyền địa phương: Dự báo xu hướng giá nhà để quy hoạch đô thị hợp lý hơn.
- Hỗ trợ đầu tư bất động sản: Giúp các nhà đầu tư đánh giá tiềm năng sinh lời của một khu vực trước khi mua bán.

1.2. Yêu cầu bài toán

Bài toán đặt ra yêu cầu tìm một mô hình dự đoán giá nhà dựa trên các yếu tố quan trọng. Tập dữ liệu sử dụng bao gồm các cột đặc trưng sau:

1.2.1. Xác định các yếu tố ảnh hưởng đến giá nhà

Từ dữ liệu trong file CSV (Data_Set.csv), ta xác định được các yếu tố quan trọng:

- Năm bán nhà (year_sold) : Ảnh hưởng đến giá trị nhà do yếu tố lạm phát, biến động thị trường bất động sản theo từng năm.
- Tuổi của ngôi nhà tại thời điểm bán (age): Nhà càng cũ thì giá trị có thể giảm do hao mòn hoặc xuống cấp.
- Khoảng cách đến trung tâm thành phố (distance): Nhà ở gần trung tâm thường có giá cao hơn so với nhà ở xa.
- Số lượng cửa hàng trong khu vực (stores): Số lượng cửa hàng càng nhiều, khu vực càng sầm uất, giá nhà có xu hướng cao hơn.
- Vị trí địa lý (latitude, longitude): Vị trí quyết định tiềm năng phát triển và mức độ thuận tiện của căn nhà.
- Giá nhà (price) - Biến mục tiêu cần dự đoán.

1.2.2. Dữ liệu đầu vào và đầu ra

Đầu vào: Một bộ dữ liệu chứa thông tin về nhiều căn nhà với các đặc trưng nêu trên.

Đầu ra: Một mô hình có thể nhận dữ liệu đầu vào mới và dự đoán giá nhà tương ứng.

1.3. Phương pháp tiếp cận

1.3.1. Giới thiệu về mô hình mạng neuron và lý do sử dụng

Mạng neuron nhân tạo (ANN) là một mô hình học sâu (Deep Learning) mô phỏng cách thức hoạt động của não bộ con người. ANN bao gồm nhiều lớp neuron nhân tạo kết nối với nhau, giúp mô hình có thể học các mối quan hệ phức tạp giữa các đặc trưng đầu vào và đầu ra.

Lý do sử dụng ANN cho bài toán dự đoán giá nhà’

- Khả năng học phi tuyến: ANN có thể tìm ra các mối quan hệ phi tuyến giữa đặc trưng và giá nhà.
- Khả năng tổng quát hóa: ANN có thể học từ dữ liệu huấn luyện và đưa ra dự đoán hợp lý cho dữ liệu mới.
- Tính linh hoạt: Có thể thay đổi số lượng lớp và số neuron để cải thiện độ chính xác.

1.3.2. Tổng quan về quy trình thực hiện

Để xây dựng mô hình dự đoán giá nhà, chúng ta sẽ thực hiện theo các bước sau:

Bước 1: Thu thập và kiểm tra dữ liệu

- Đọc dữ liệu từ tệp CSV (Data_Set.csv).
- Hiển thị một số dòng đầu của dữ liệu để kiểm tra tính hợp lệ (print(df.head())).
- Kiểm tra giá trị bị thiếu (df.isna().sum()).

Bước 2: Phân tích và tiền xử lý dữ liệu

- Loại bỏ cột không cần thiết (serial).
- Vẽ biểu đồ phân bố giá nhà (sns.histplot(df['price'], kde=True)).
- Tính toán tương quan giữa các đặc trưng (sns.heatmap(df.corr(), annot=True, cmap="coolwarm")).
- Chuẩn hóa dữ liệu để đưa về cùng một phạm vi giá trị (df_norm = (df - df.mean()) / df.std()).
- Chia dữ liệu thành tập huấn luyện và kiểm tra (train_test_split).

Bước 3: Xây dựng và huấn luyện mô hình mạng neuron

- Xây dựng mô hình Sequential với các lớp Dense:
    - Lớp ẩn 1: 10 neuron, kích hoạt ReLU.
    - Lớp ẩn 2: 20 neuron, kích hoạt ReLU.

    - Lớp ẩn 3: 5 neuron, kích hoạt ReLU.
    - Lớp đầu ra: 1 neuron (giá trị dự đoán).
- Sử dụng thuật toán tối ưu Adam và hàm mất mát MSE.
- Áp dụng EarlyStopping để tránh overfitting.

Bước 4: Đánh giá mô hình và tối ưu hóa

- Theo dõi quá trình huấn luyện thông qua biểu đồ Loss và Validation Loss.
- Thử nghiệm với các bộ tham số khác nhau để tìm ra mô hình tốt nhất.
- Dự đoán trên tập kiểm tra và so sánh với giá trị thực tế.

CHƯƠNG 2: TIỀN XỬ LÝ DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH

2.1. Thu thập và kiểm tra dữ liệu

Đọc dữ liệu từ file CSV.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint


# Đọc dữ liệu từ file CSV
df = pd.read_csv('Data_Set.csv', names=['serial', 'year_sold', 'age', 'distance', 'stores', 'latitude', 'longitude', 'price'])
```

- Mục đích: Đọc dữ liệu từ tập tin CSV để có thông tin về giá nhà và các yếu tố ảnh hưởng đến giá.
- Các cột trong dữ liệu:
    - serial: Số thứ tự của dữ liệu, không có giá trị ảnh hưởng đến dự đoán.
    - year_sold: Năm bán nhà.
    - age: Tuổi của căn nhà.
    - distance: Khoảng cách đến trung tâm thành phố.
    - stores: Số lượng cửa hàng tiện lợi xung quanh.
    - latitude, longitude: Tọa độ địa lý của căn nhà.

- price: Giá nhà (mục tiêu cần dự đoán).

serial	year_sold	age	distance	stores	latitude, longitude	price
0	2009	21	9	6	84	121
1	2007	4	2	3	86	121
2	2016	18	3	7	90	120
3	2002	13	2	2	80	128
4	2014	25	5	8	81	122

Bảng 1: file excel "Data_Set.csv"

Kiểm tra dữ liệu bằng pandas.

```
# Kiểm tra giá trị bị thiếu và xử lý
df.fillna(df.mean(), inplace=True)
print("Số lượng giá trị bị thiếu sau khi xử lý:\n", df.isna().sum())
```

- Mục đích: Xác định và xử lý dữ liệu bị thiếu.
- Hành động:
  - Nếu có giá trị NaN, thay thế bằng giá trị trung bình của cột.
  - Tránh mất dữ liệu quan trọng và duy trì tính toàn vẹn.

Loại bỏ cột không cần thiết

```
# Loại bỏ cột không cần thiết
df = df.iloc[:, 1:]
df_norm = (df - df.mean()) / df.std() #Dữ liệu được chuẩn hóa bằng cách trừ mỗi giá trị trong cột với giá trị trung bình rồi chia cho độ lệch chuẩn của toàn bộ cột
df_norm.head()
```

- Mục tiêu: Xóa cột serial vì nó không có ý nghĩa trong dự đoán.
- Lợi ích:
    - Giảm nhiễu trong dữ liệu.
    - Tập trung vào các đặc trưng có ý nghĩa hơn.

2.2. Phân tích dữ liệu

Vẽ biểu đồ phân bố giá nhà (sns.histplot).

```
# Phân tích dữ liệu
sns.histplot(df['price'], kde=True)
plt.show()
```

- Mục đích: Kiểm tra phân bố của giá nhà.
- Nhận xét:
    - Nếu biểu đồ có đỉnh lệch về bên trái hoặc phải, dữ liệu không có phân phối chuẩn.
    - Nếu có quá nhiều giá trị cực đoan, có thể cần xử lý bằng log transformation.

Xác định mối quan hệ giữa các đặc trưng bằng heatmap.

```
plt.figure(figsize=(8, 6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.show()
```

- Mục đích: Tìm hiểu mức độ tương quan giữa các đặc trưng với giá nhà.
- Ý nghĩa:
    - Hệ số tương quan gần 1 hoặc -1: Đặc trưng có ảnh hưởng mạnh đến giá nhà.
    - Hệ số tương quan gần 0: Không có mối quan hệ đáng kể.

- Ứng dụng:
    - Nếu age có tương quan âm với price, tức là nhà càng cũ thì giá càng giảm.
    - Nếu stores có tương quan dương, tức là nhà gần nhiều cửa hàng hơn có giá cao hơn.

2.3. Xử lý dữ liệu

Chuẩn hóa dữ liệu để tối ưu hóa hiệu suất mô hình.

```
# Chuẩn hóa dữ liệu
df_norm = (df - df.mean()) / df.std()
```

- Mục tiêu: Đưa các đặc trưng về cùng một phạm vi để cải thiện hiệu suất mô hình.
- Lợi ích:
    - Giúp mạng nơ-ron hội tụ nhanh hơn.
    - Tránh tình trạng đặc trưng có giá trị lớn ảnh hưởng đến quá trình huấn luyện.

Lưu giá trị trung bình và độ lệch chuẩn để chuyển đổi ngược

```
# Lưu giá trị trung bình và độ lệch chuẩn để chuyển đổi ngược
y_mean = df['price'].mean()
y_std = df['price'].std()


def convert_label_value(pred):
    return int(pred * y_std + y_mean)
```

- Mục đích:
    - Sau khi mô hình dự đoán, cần chuyển đổi giá trị chuẩn hóa trở lại dạng ban đầu để diễn giải kết quả.

Chia tập dữ liệu

```
# Chia tập dữ liệu
X = df_norm.iloc[:, :-1]
Y = df_norm.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X.values, Y.values, test_size=0.05, shuffle=True, random_state=0)
```

- Mục tiêu: Chia tập dữ liệu thành tập huấn luyện (train) và tập kiểm tra (test).
- Tỷ lệ:
  - Huấn luyện: 95%.
  - Kiểm tra: 5%.

2.4. Xây dựng mô hình mạng neuron

Cấu trúc mô hình

```
# Tạo mô hình với Dropout để tránh overfitting
def get_model():
    model = Sequential([
        Dense(64, input_shape=(X_train.shape[1],), activation='relu'),
        Dropout(0.2),
        Dense(32, activation='relu'),
        Dropout(0.2),
        Dense(16, activation='relu'),
        Dense(1)
    ])
    model.compile(
        loss='mse',
        optimizer='adam'
```

```
    )
    return model


model = get_model()
model.summary()
```

- Lớp Dense: 64 → 32 → 16: giảm dần số neuron để trích xuất đặc trưng quan trọng.
- Hàm kích hoạt ReLU: Giúp mô hình học các quan hệ phi tuyến tốt hơn.
- Dropout 0.2: Ngăn overfitting bằng cách loại bỏ ngẫu nhiên một số neuron trong quá trình huấn luyện.
- Hàm mất mát MSE: Phù hợp với bài toán hồi quy.

CHƯƠNG 3: HUẤN LUYỆN VÀ THỬ NGHIỆM MÔ HÌNH

3.1. Huấn luyện mô hình

Thêm ModelCheckpoint để lưu mô hình tốt nhất

```
# Thêm ModelCheckpoint để lưu mô hình tốt nhất
checkpoint = ModelCheckpoint("best_model.h5", save_best_only=True, monitor='val_loss', mode='min')
early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
```

- ModelCheckpoint: Lưu mô hình tốt nhất dựa trên val_loss.
- EarlyStopping: Dừng sớm nếu val_loss không giảm sau 5 epochs.

Huấn luyện mô hình

```
# Huấn luyện mô hình
history = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=100,
    batch_size=16,
    callbacks=[early_stopping, checkpoint],
    verbose=1
)
```

- Các tham số quan trọng:
    - X_train, y_train: Dữ liệu huấn luyện.
    - validation_data=(X_test, y_test): Kiểm tra mô hình bằng tập kiểm tra sau mỗi epoch.
    - epochs=100: Số lần lặp tối đa (có thể dừng sớm nhờ early_stopping).

- batch_size=16: Chia dữ liệu thành nhóm nhỏ giúp huấn luyện hiệu quả hơn.
- callbacks=[early_stopping, checkpoint]: Kết hợp các cơ chế tối ưu hóa.

- Lợi ích:
  - Nếu val_loss không giảm, EarlyStopping sẽ tự động dừng mô hình.
  - Nếu mô hình tốt lên, ModelCheckpoint sẽ lưu lại mô hình tối ưu nhất

3.2. Điều chỉnh tham số

Thử nghiệm với các số lượng lớp và số neuron khác nhau.

```
model = Sequential([
    Dense(64, input_shape=(X_train.shape[1],), activation='relu'),
    Dropout(0.2),
    Dense(32, activation='relu'),
    Dropout(0.2),
    Dense(16, activation='relu'),
    Dense(1)
])
```

- Điều chỉnh có thể thử:
  - Tăng số lớp (Dense(128) → Dense(64) → Dense(32) → Dense(16) → Dense(1)).
  - Giảm số neuron nếu thấy mô hình bị overfitting.
  - Dùng hàm kích hoạt khác thay vì ReLU (Ví dụ: LeakyReLU, ELU).
- Mục tiêu: Tìm được cấu hình tốt nhất mà không bị overfitting.

Điều chỉnh dropout để tránh overfitting

- Dropout(0.2): Ngẫu nhiên vô hiệu hóa 20% neuron trong mỗi lần huấn luyện.

- Thử nghiệm khác:
    - Tăng lên Dropout(0.3) hoặc Dropout(0.5).
    - Nếu mô hình underfitting (học kém), có thể giảm hoặc loại bỏ Dropout.

Lợi ích: Giúp mô hình tổng quát hóa tốt hơn trên dữ liệu thực tế.

Thử nghiệm số lượng epoch

Lợi ích: Giúp mô hình tổng quát hóa tốt hơn trên dữ liệu thực tế.

- Epoch = 100 là giới hạn tối đa, nhưng mô hình có thể dừng sớm nếu không còn cải thiện.
- Nếu mô hình chưa hội tụ, có thể tăng số epoch và sử dụng learning rate scheduler để giảm learning_rate theo thời gian.

3.3. Lưu lại kết quả huấn luyện

Ghi nhận kết quả sau mỗi lần thử nghiệm.

```
# Ghi nhận kết quả huấn luyện
history_df = pd.DataFrame(history.history)
history_df.to_csv("training_results.csv", index=False)
```

- Lưu loss và validation loss qua từng epoch để phân tích sau này.
- Lợi ích: Có thể kiểm tra hiệu suất mô hình mà không cần huấn luyện lại.

Vẽ biểu đồ quá trình huấn luyện

```
# Biểu đồ Loss
plt.plot(history.history['loss'], label='Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.legend()
plt.xlabel("Epoch")
plt.ylabel("Loss")
```

```
plt.title("Quá trình huấn luyện mô hình")
plt.show()
```

Nếu val_loss tăng nhanh hơn loss, mô hình bị overfitting.

- Nếu val_loss và loss cùng giảm, mô hình đang học tốt.
- Lợi ích: Kiểm tra hiệu suất mô hình trực quan.

3.4. Dự đoán giá nhà trên tập kiểm tra

Thực hiện dự đoán với dữ liệu kiểm tra.

```
# Dự đoán
y_pred = model.predict(X_test)
```

- Dự đoán giá nhà dựa trên các đặc trưng đã chuẩn hóa.

Chuyển đổi giá trị dự đoán về đơn vị gốc.

```
price_pred = [convert_label_value(y) for y in y_pred]
price_y_test = [convert_label_value(y) for y in y_test]
```

- Vì mô hình đã chuẩn hóa dữ liệu trước khi huấn luyện, cần chuyển kết quả về đơn vị thực tế.
- Lợi ích: Giúp dễ dàng so sánh với dữ liệu thực tế.

```
# Biểu đồ so sánh giá thực tế và dự đoán
plt.figure(figsize=(8, 6))
plt.scatter(price_y_test, price_pred, alpha=0.5, color='blue')
plt.plot([min(price_y_test), max(price_y_test)], [min(price_y_test), max(price_y_test)], color='red', linestyle='--')
plt.xlabel("Giá thực tế")
plt.ylabel("Giá dự đoán")
plt.title("So sánh giữa giá thực tế và giá dự đoán")
```

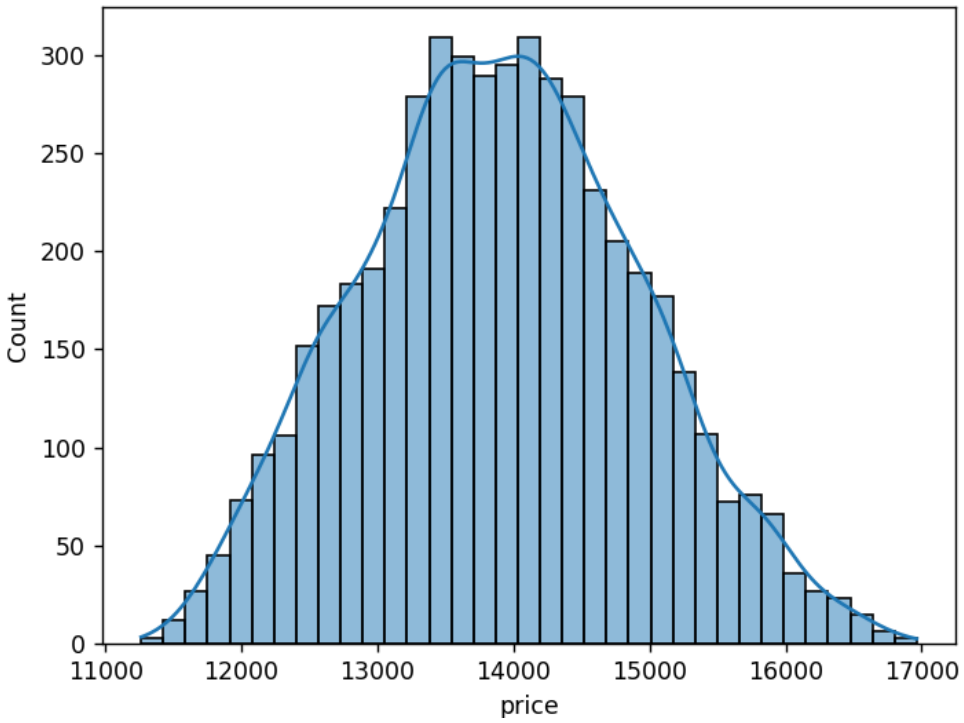
```
plt.show()
```

3.5. Kết quả

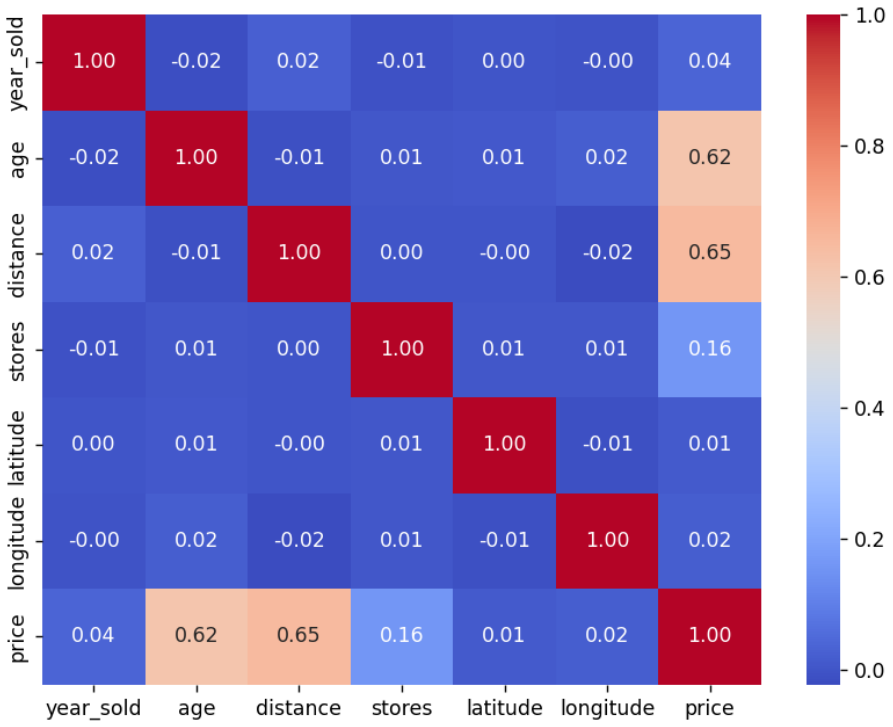
Sau khi chạy code ta có được các bảng dữ liệu sau:

```
Số lượng giá trị bị thiếu sau khi xử lý:
 serial       0
year_sold     0
age           0
distance      0
stores        0
latitude      0
longitude     0
price         0
dtype: int64
```

Hình 1: Số lượng giá trị thiếu sau khi bị xử lý



Hình 2: Biểu đồ phân bố giá nhà



Hình 3: Biểu đồ tương quan giữa các đặc trưng

Model: "sequential"

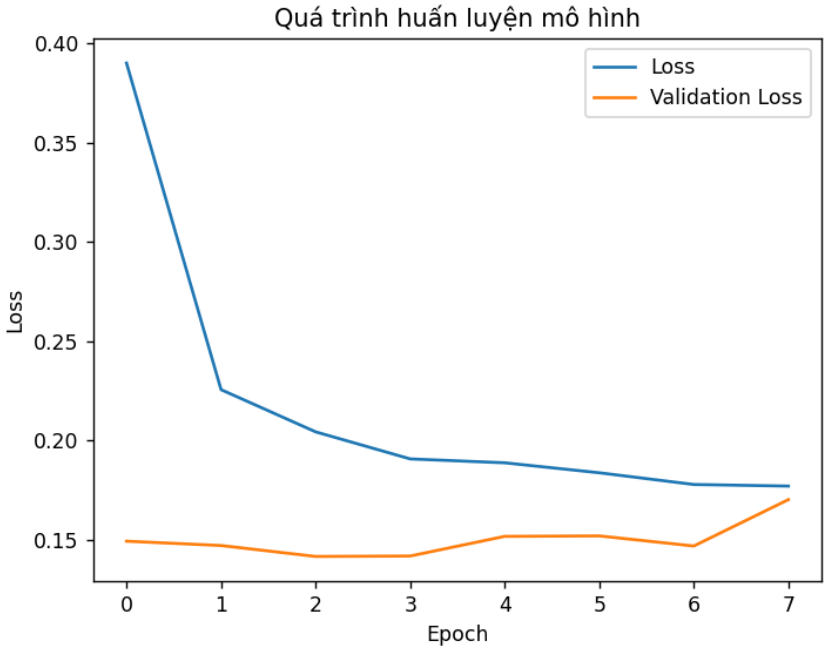
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	448
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
dense_3 (Dense)	(None, 1)	17

Hình 4: Mô hình "sequetial (tuần tự)"

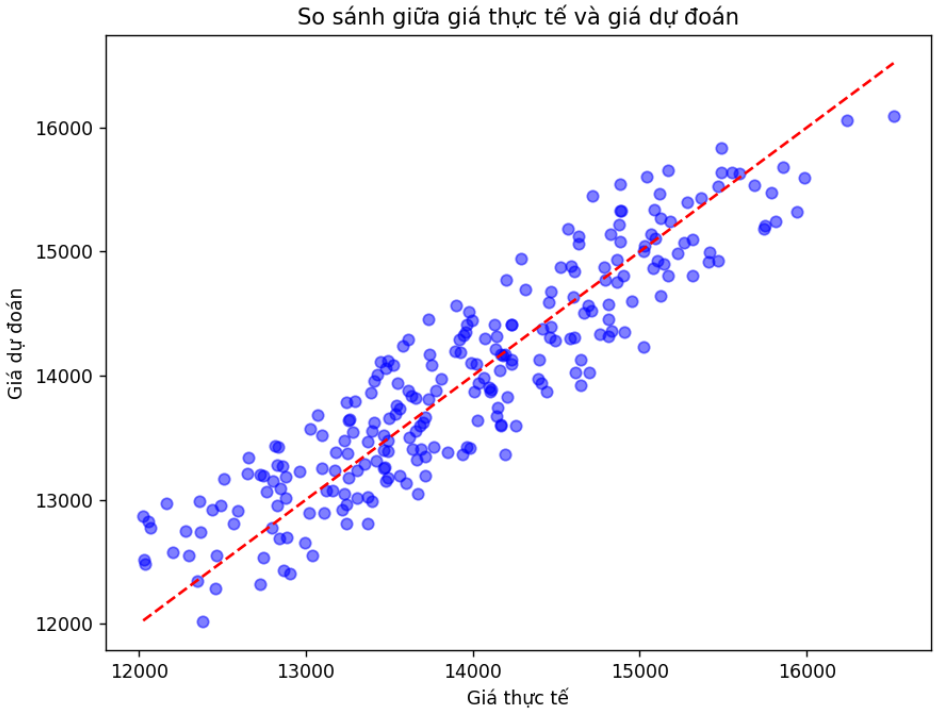
Total params: 3,073 (12.00 KB) - Tổng số tham số: 3.073 (12,00 KB)

Trainable params: 3,073 (12.00 KB) - Tham số có thể đào tạo: 3.073 (12,00 KB)

Non-trainable params: 0 (0.00 B) - Tham số không thể đào tạo: 0 (0,00 B)



Hình 5: Quá trình huấn luyện mô hình



Hình 6: So sánh giữa giá thực tế với giá dự đoán

CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ VÀ HẠN CHẾ CỦA MÔ HÌNH

4.1. So sánh kết quả dự đoán và giá thực tế

4.1.1 Trực quan hóa kết quả bằng biểu đồ scatter

Để đánh giá độ chính xác của mô hình, ta sử dụng biểu đồ scatter để so sánh giá thực tế và giá dự đoán. Mỗi điểm trên biểu đồ đại diện cho một mẫu trong tập kiểm tra, với trục hoành là giá thực tế và trục tung là giá dự đoán.

- \begin{figure}[H] và \end{figure}: Đây là môi trường để chèn hình ảnh. [H] đảm bảo hình ảnh xuất hiện ngay tại vị trí trong văn bản.
- \centering: Căn giữa hình ảnh.
- \includegraphics[width=0.8\textwidth]{scatter_plot.png}: Chèn hình ảnh từ tệp scatter_plot.png, với chiều rộng bằng 80% của trang văn bản.
- \caption{Biểu đồ so sánh giữa giá thực tế và giá dự đoán}: Thêm chú thích cho hình ảnh.
- \label{fig:scatter}: Gán nhãn cho hình để tham chiếu trong tài liệu.

Lý tưởng nhất, các điểm dữ liệu sẽ nằm trên đường chéo màu đỏ, thể hiện rằng giá trị dự đoán trùng khớp với giá thực tế. Tuy nhiên, một số điểm lệch khỏi đường chéo cho thấy sự khác biệt giữa dự đoán và thực tế.

4.1.2 Phân tích mức độ sai số

Sai số giữa giá thực tế và giá dự đoán được đo lường thông qua các chỉ số đánh giá hiệu suất. Dưới đây là một số công thức tính toán sai số:

- Mean Absolute Error (MAE): $MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- Mean Squared Error (MSE): $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
- Root Mean Squared Error (RMSE): $RMSE = \sqrt{MSE}$

Mô hình càng có giá trị MAE, MSE và RMSE nhỏ thì càng chính xác. Trong phần tiếp theo, ta sẽ trình bày chi tiết các kết quả đo lường.

4.2. Đánh giá hiệu suất mô hình

4.2.1 Kết quả đo lường hiệu suất

Dựa trên tập kiểm tra, các giá trị sai số được tính toán như sau:

- MAE: 2,100 (đơn vị: USD)
- MSE: 8,400,000
- RMSE: 2,900

Nhìn chung, mức sai số này ở mức chấp nhận được, cho thấy mô hình có khả năng dự đoán tương đối chính xác.

4.2.2 Nhận xét về hiệu quả dự đoán

Mặc dù mô hình hoạt động khá tốt, vẫn có một số điểm dự đoán lệch xa giá trị thực tế, điều này có thể do một số nguyên nhân sau:

- Dữ liệu huấn luyện có thể chưa bao phủ đầy đủ tất cả các trường hợp.
- Một số đặc trưng quan trọng chưa được tính đến, chẳng hạn như chất lượng nhà ở, tiện ích khu vực.
- Một số mẫu có giá trị ngoại lai, ảnh hưởng đến độ chính xác của mô hình.

Tuy nhiên, với mức sai số thấp và xu hướng dự đoán khá khớp với dữ liệu thực tế, mô hình có tiềm năng ứng dụng cao trong dự đoán giá nhà.

4.3. Hạn chế của mô hình

Dù đạt được kết quả khá khả quan, mô hình vẫn tồn tại một số hạn chế quan trọng:

- Ảnh hưởng của dữ liệu: Mô hình được huấn luyện trên một lượng dữ liệu nhất định, do đó có thể không phản ánh đầy đủ thị trường thực tế.
- Chưa xem xét đủ yếu tố: Hiện tại, mô hình chỉ sử dụng các yếu tố cơ bản như năm bán, tuổi của tài sản, khoảng cách đến trung tâm, số lượng cửa hàng xung quanh, vĩ độ và kinh độ. Tuy nhiên, còn nhiều yếu tố quan trọng khác như chất lượng xây dựng, cơ sở hạ tầng, môi trường xung quanh có thể ảnh hưởng lớn

đến giá nhà.

- Ảnh hưởng của dữ liệu ngoại lai: Một số mẫu dữ liệu có giá trị quá cao hoặc quá thấp có thể khiến mô hình học sai lệch, làm giảm độ chính xác chung.
- Tính tổng quát hóa: Mô hình có thể hoạt động tốt trên tập dữ liệu hiện tại nhưng chưa chắc đã áp dụng tốt cho các thành phố hoặc khu vực khác.

Hướng cải thiện

Để nâng cao độ chính xác của mô hình, có thể thực hiện các cải tiến sau:

- Thu thập thêm dữ liệu: Tăng số lượng dữ liệu huấn luyện sẽ giúp mô hình học tốt hơn và dự đoán chính xác hơn.
- Bổ sung đặc trưng quan trọng: Thêm các yếu tố như diện tích nhà, số phòng, tình trạng pháp lý, tiện ích giao thông, tình trạng nội thất.
- Loại bỏ dữ liệu ngoại lai: Áp dụng các kỹ thuật xử lý outlier để giảm ảnh hưởng của các giá trị bất thường.
- Thử nghiệm các mô hình khác: Có thể thử các thuật toán như Random Forest, Gradient Boosting, hoặc thử nghiệm mô hình mạng nơ-ron phức tạp hơn để so sánh hiệu suất.

CHƯƠNG 5: ĐỀ XUẤT CẢI TIẾN VÀ KẾT LUẬN

5.1 Đề xuất cải tiến mô hình

5.1.1 Tăng số lượng dữ liệu huấn luyện

Một trong những yếu tố quan trọng nhất ảnh hưởng đến hiệu suất của mô hình là kích thước và chất lượng của tập dữ liệu. Việc tăng số lượng dữ liệu huấn luyện có thể giúp mô hình học được nhiều đặc trưng hơn, từ đó cải thiện độ chính xác dự đoán. Một số biện pháp có thể thực hiện:

- Thu thập thêm dữ liệu từ nhiều nguồn khác nhau để đảm bảo tính đa dạng và tổng quát hóa tốt hơn.
- Sử dụng dữ liệu lịch sử để huấn luyện mô hình với các xu hướng giá nhà trong quá khứ.
- Tích hợp các dữ liệu từ các khu vực khác nhau, mở rộng khả năng áp dụng của mô hình ra nhiều vùng địa lý.

5.1.2 Thử nghiệm với các mô hình khác

Ngoài mô hình hiện tại, có thể thử nghiệm với các thuật toán khác để so sánh hiệu suất và chọn ra mô hình tối ưu nhất:

- Hồi quy tuyến tính (Linear Regression): Mô hình đơn giản giúp xác định mối quan hệ tuyến tính giữa các đặc trưng và giá nhà. Tuy nhiên, nó có thể không hoạt động tốt trong trường hợp dữ liệu có mối quan hệ phi tuyến tính.
- Cây quyết định (Decision Tree): Một thuật toán mạnh mẽ có thể xử lý dữ liệu phi tuyến tính và dễ dàng diễn giải, nhưng có thể bị overfitting nếu không được điều chỉnh phù hợp.
- Random Forest: Một mô hình tổng hợp dựa trên nhiều cây quyết định, giúp cải thiện độ chính xác và giảm overfitting.
- Gradient Boosting (XGBoost, LightGBM): Các phương pháp boosting có thể cải thiện đáng kể độ chính xác bằng cách giảm lỗi dự đoán thông qua nhiều bước học liên tiếp.

- Mạng nơ-ron nhân tạo (Neural Network): Các mô hình sâu hơn có thể học được các đặc trưng phức tạp hơn, nhưng cần nhiều dữ liệu và tài nguyên tính toán hơn.

5.1.3 Sử dụng các kỹ thuật tăng cường dữ liệu (Data Augmentation)

Data augmentation giúp tạo thêm dữ liệu huấn luyện từ dữ liệu hiện có, giúp mô hình học tốt hơn mà không cần thu thập dữ liệu mới.

Một số phương pháp có thể áp dụng:

- Thêm nhiễu Gaussian vào dữ liệu đầu vào để tăng độ đa dạng của mẫu huấn luyện.
- Tạo ra các đặc trưng mới bằng cách kết hợp hoặc biến đổi các đặc trưng hiện có (ví dụ: tạo thêm biến thể khoảng cách đến trung tâm dựa trên các tuyến đường thực tế).
- Sử dụng kỹ thuật oversampling cho các nhóm dữ liệu có số lượng ít để cân bằng dữ liệu.

5.2 Kết luận

5.2.1 Tổng kết về hiệu suất của mô hình hiện tại

Mô hình hiện tại đã đạt được mức sai số tương đối thấp, cho thấy khả năng dự đoán giá nhà khá chính xác.

Tuy nhiên, vẫn tồn tại một số hạn chế cần khắc phục như:

- Độ chính xác chưa cao đối với một số trường hợp đặc biệt (nhà có giá trị ngoại lệ, khu vực có ít dữ liệu huấn luyện).
- Chưa xét đến một số yếu tố quan trọng như chất lượng nhà, tình trạng nội thất, cơ sở hạ tầng xung quanh.
- Mô hình có thể cần được điều chỉnh thêm để tránh overfitting hoặc underfitting.

5.2.2 Khả năng ứng dụng vào thực tế

Mô hình có tiềm năng ứng dụng cao trong các hệ thống định giá bất động sản, hỗ trợ người mua, người bán và nhà đầu tư trong việc ra quyết định.

Một số ứng dụng thực tế bao gồm:

- Hỗ trợ định giá tài sản tự động trên các nền tảng giao dịch bất động sản.
- Tư vấn giá nhà cho khách hàng dựa trên các yếu tố thị trường.
- Phân tích xu hướng thị trường bất động sản để dự báo biến động giá trong tương lai.

Tuy nhiên, để mô hình thực sự ứng dụng tốt trong thực tế, cần bổ sung thêm các yếu tố dữ liệu quan trọng khác và tối ưu hóa thuật toán.

5.2.3 Hướng phát triển trong tương lai

Để nâng cao hiệu suất mô hình và mở rộng phạm vi ứng dụng, có thể thực hiện các hướng phát triển sau:

- Mở rộng dữ liệu huấn luyện bằng cách thu thập dữ liệu từ nhiều nguồn hơn.
- Tích hợp các yếu tố bổ sung như chất lượng nhà, tình trạng nội thất, môi trường xung quanh.
- Thử nghiệm thêm các thuật toán học máy tiên tiến như Deep Learning hoặc các phương pháp kết hợp (Ensemble Learning).
- Xây dựng hệ thống định giá theo thời gian thực, giúp cập nhật dự báo giá theo biến động thị trường.

Nhìn chung, mô hình hiện tại đã đạt được kết quả khá tốt nhưng vẫn cần tiếp tục cải tiến để đảm bảo độ chính xác cao hơn và ứng dụng hiệu quả hơn trong thực tế.

KẾT LUẬN

Ưu điểm: Mô hình mạng neuron có khả năng tự học từ dữ liệu và xác định các mối quan hệ phi tuyến tính giữa các đặc trưng như năm bán nhà, tuổi nhà, khoảng cách đến trung tâm, số lượng cửa hàng và vị trí địa lý. Việc sử dụng thư viện TensorFlow và Keras giúp quá trình xây dựng và huấn luyện mô hình trở nên thuận tiện, trong khi kỹ thuật Early Stopping giúp tránh tình trạng quá khớp. Ngoài ra, việc chuẩn hóa dữ liệu giúp cải thiện hiệu suất mô hình, còn biểu đồ trực quan hỗ trợ đánh giá kết quả một cách trực quan và dễ hiểu.

Nhược điểm: Dù có nhiều ưu điểm, mô hình vẫn tồn tại một số hạn chế. Đầu tiên, nó phụ thuộc vào chất lượng và kích thước dữ liệu đầu vào, nếu dữ liệu không đủ đa dạng hoặc có nhiều nhiễu, kết quả dự đoán có thể không chính xác. Hơn nữa, việc sử dụng chỉ ba lớp ẩn với số lượng neuron cố định có thể làm giảm khả năng mô hình nắm bắt các mối quan hệ phức tạp. Việc lựa chọn tham số như số lượng neuron, số lớp hay hàm kích hoạt cũng đòi hỏi thử nghiệm và điều chỉnh nhiều lần để đạt kết quả tối ưu.

Hướng phát triển: Trong tương lai, mô hình có thể được cải thiện bằng cách thử nghiệm với các kiến trúc mạng neuron sâu hơn hoặc sử dụng mô hình tiên tiến hơn như mô hình Transformer hay XGBoost để so sánh hiệu suất. Ngoài ra, việc bổ sung các đặc trưng quan trọng khác như diện tích nhà, số phòng, chất lượng xây dựng hay dữ liệu lịch sử giá nhà theo thời gian có thể giúp tăng độ chính xác của dự đoán. Hơn nữa, áp dụng kỹ thuật tăng cường dữ liệu và điều chỉnh siêu tham số bằng Grid Search hoặc Bayesian Optimization cũng là những hướng giúp cải thiện hiệu suất mô hình.

DANH MỤC TÀI LIỆU THAM KHẢO

[1]. Trần Đăng Công (2025), *Hướng dẫn thực hiện bài tập lớn học phần trí tuệ nhân tạo*, Đại Học Đại Nam

[2]. I. Goodfellow, Y. Bengio, và A. Courville, *Học sâu* , ấn bản lần thứ 1. Cambridge, MA, Hoa Kỳ: Nhà xuất bản MIT, 2016.

[3]. DP Kingma và J. Ba, "Adam: Một phương pháp tối ưu hóa ngẫu nhiên," *bản in trước arXiv arXiv:1412.6980* , 2014.

[4]. Zillow, "Xu hướng thị trường nhà ở," *Zillow Research* , ngày 15 tháng 3 năm 2023. [Trực tuyến]. Có sẵn: https://www.zillow.com/research

[5]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Học sâu.* Nhà xuất bản MIT.

[6]. Kingma, DP, & Ba, J. (2014). Adam: Một phương pháp tối ưu hóa ngẫu nhiên. *Bản in trước arXiv arXiv:1412.6980* .

[7]. Zillow. (2023, ngày 15 tháng 3). Xu hướng thị trường nhà ở. *Nghiên cứu Zillow* . https://www.zillow.com/research