SIG720 Task 5D

BUSINESS REPORT

ON

Housing Price Prediction Using Regression Models

(Based on the Paris Housing Dataset)

Submitted By: Sahid

MS Data Science

Date: Aug 03, 2025

Table of Contents

List of tables

List of Figures

1. Introduction

1.1 Objective of the Study

The primary objective of this study is to develop machine learning regression models that can accurately predict house prices using structured property data. This includes evaluating and comparing model performance using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score. The project aims to not only build a predictive model but also interpret its outputs through feature importance and deploy it as an interactive web application.

1.2 Dataset Source

As per the updated guidelines by **Deakin University**, the dataset used is the **Paris Housing Price Prediction** dataset, sourced from Kaggle. This dataset consists of synthetically generated housing data that resembles real-world pricing behaviors and includes 10,000 entries with 17 attributes related to property features and sale price.

---

2. Dataset Overview

2.1 Dataset Shape & Features

The dataset has **10,000 rows** and **17 columns**, with the target variable being price. The feature set includes:

- **Numerical Variables:** squareMeters, numberOfRooms, floors, basement, attic, garage, etc.
- **Binary/Categorical Variables:** hasYard, hasPool, isNewBuilt, hasStormProtector, hasStorageRoom, etc.
- **Temporal/Location-Based:** made (year of construction), cityCode, cityPartRange, numPrevOwners.

squareMeters	numberOfRooms	hasYard	hasPool	floors	cityCode	cityPartRange	numPrevOwners	made	isNewBuilt	hasStormProte	basement	attic	garage	hasStorageRoo	hasGuestRoom	price
75523	3	0	1	63	9373	3	8	2005	0	1	4313	9005	956	0	7	7,559,081.50
80771	39	1	1	98	39381	8	6	2015	1	0	3653	2436	128	1	2	8,085,989.50
55712	58	0	1	19	34457	6	8	2021	0	0	2937	8852	135	1	9	5,574,642.10
32316	47	0	0	6	27939	10	4	2012	0	1	659	7141	359	0	3	3,232,561.20
70429	19	1	1	90	38045	3	7	1990	1	0	8435	2429	292	1	4	7,055,052.00

(Table 1 : First 5 rows from the dataset)

2.2 Descriptive Statistics

A statistical summary of the dataset provides an overview of data distribution, central tendencies, and variability:

- The average house size is **~49,870 sq. meters**, with 50 rooms on average.
- Prices range from **₹10,313.50** to **₹10,006,770**, with a mean of **₹4,993,448**.
- Some variables like squareMeters, numberOfRooms, and garage show a wide spread and potential outliers.

	squareMeters	numberOfRooms	hasYard	hasPool	floors	cityCode	cityPartRange	numPrevOwners	made	isNewBuilt	hasStormProtector	basement	attic	garage	hasStorageRoom	hasGuestRoom	price
count	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10,000.00	1.00E+04
mean	49870.1312	50.3584	0.5087	0.4968	50.2763	50225.4861	5.5101	5.5217	2005.4885	0.4991	0.4999	5033.1039	5028.0106	553.1212	0.503	4.99	4.99E+06
std	28774.37535	28.816696	0.499949	0.500015	28.889171	29006.6758	2.872024	2.856667	9.30809	0.500024	0.500025	2876.72954	2894.33221	262.05017	0.500016	3.18	2.88E+06
min	89	1	0	0	1	3	1	1	1990	0	0	0	1	100	0	0.00	1.03E+04
25%	25098.5	25	0	0	25	24693.75	3	3	1997	0	0	2559.75	2512	327.75	0	2.00	2.52E+06
50%	50105.5	50	1	0	50	50693	5	5	2005.5	0	0	5092.5	5045	554	1	5	5.02E+06
75%	74609.75	75	1	1	76	75683.25	8	8	2014	1	1	7511.25	7540.5	777.25	1	8	7.47E+06
max	99999	100	1	1	100	99953	10	10	2021	1	1	10000	10000	1000	1	10	1.00E+07

(Table 2 : Summary statistics)

3. Data Preprocessing & Exploratory Data Analysis

3.1 Handling Categorical & Numerical Data

The dataset comprises 17 features and 10,000 observations related to housing prices in Paris. Among these features are both continuous numerical variables and binary indicators (e.g., presence of a yard or pool). As the modeling task involves predicting a continuous variable (price), preprocessing the input features is critical to ensure consistency and model readiness.

Categorical and Numerical Feature Handling

- All categorical variables are already numerically encoded (e.g., hasYard, hasPool, isNewBuilt) as binary (0 or 1).
- The remaining variables (like squareMeters, numberOfRooms, floors, etc.) are numerical in nature and ready for modeling.
- A derived feature named house_age was computed by subtracting the year of construction (made) from the current year (2025), representing how old a house is — a more interpretable variable than made.

Feature Scaling

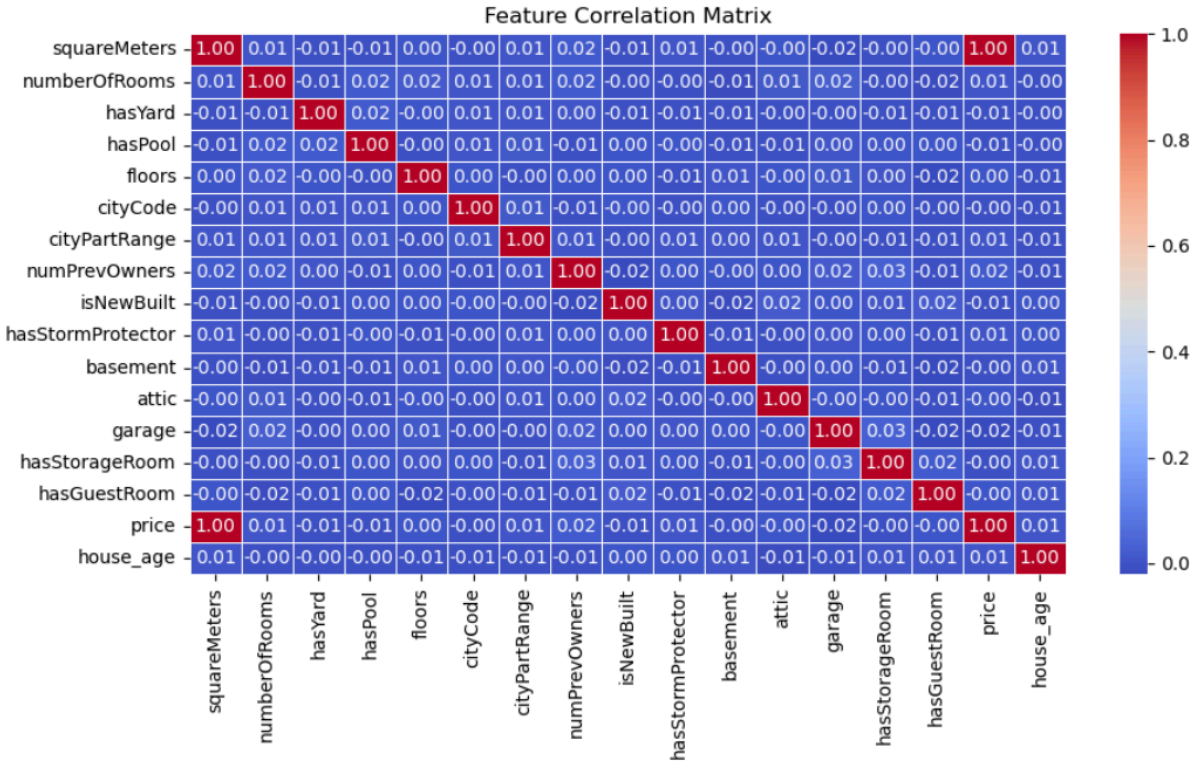
- To prepare for regression models sensitive to feature magnitudes (e.g., Linear Regression, XGBoost), all numerical features were standardized using a scaling technique.
- Standardizing ensures each feature contributes proportionally and reduces bias caused by differing units or ranges.

3.2 Visualizations & Statistical Summary

Visual exploration was carried out to better understand the distribution, spread, and relationships between features and the target variable (price). This also helps in identifying trends and potential data quality issues like outliers.

3.2.1 Correlation Heatmap

A heatmap was generated to examine linear correlations between features and with the target variable. This helps in identifying multicollinearity or strong predictive relationships.



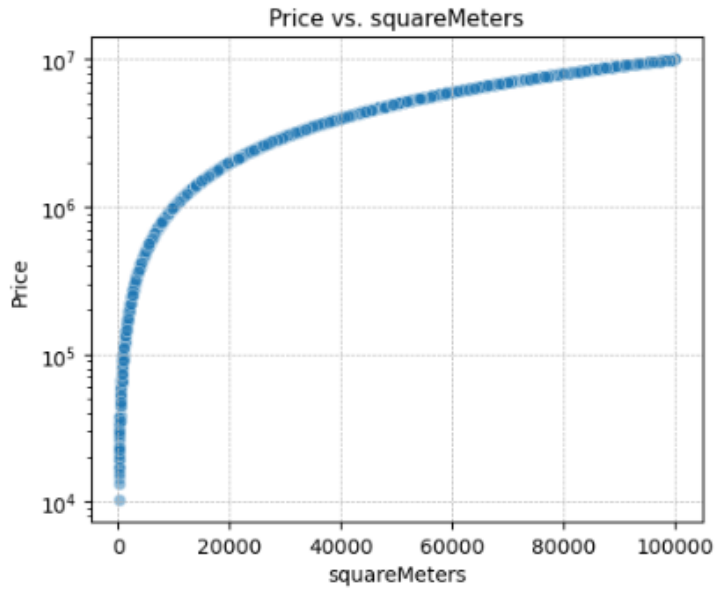
(**Figure 1: Correlation Heatmap of All Features)**

Inferences:

- squareMeters shows a **very strong positive correlation** with price, suggesting it will play a key role in prediction.
- numberOfRooms, garage, and floors also exhibit moderate positive correlation.
- Features like cityCode, attic, and basement show weak or negligible correlation with price, which may still be useful when interacting with other variables.

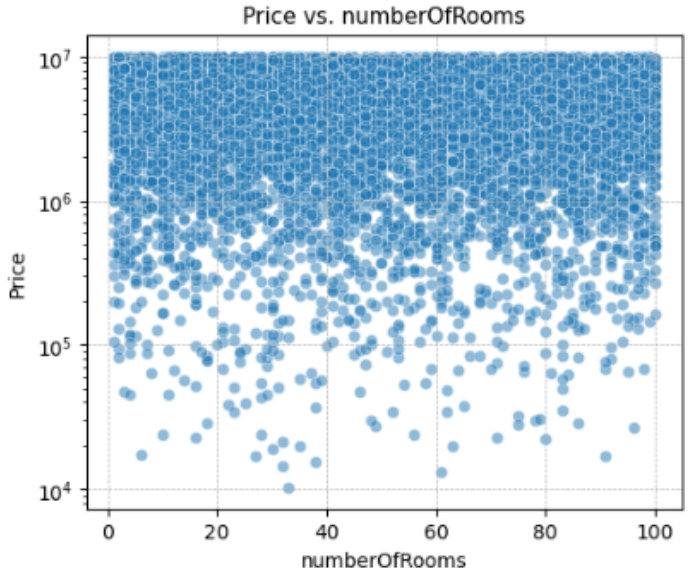
3.2.2 Scatter Plots with Price

Scatter plots between key predictors and price were used to visually assess patterns, trends, and relationships.



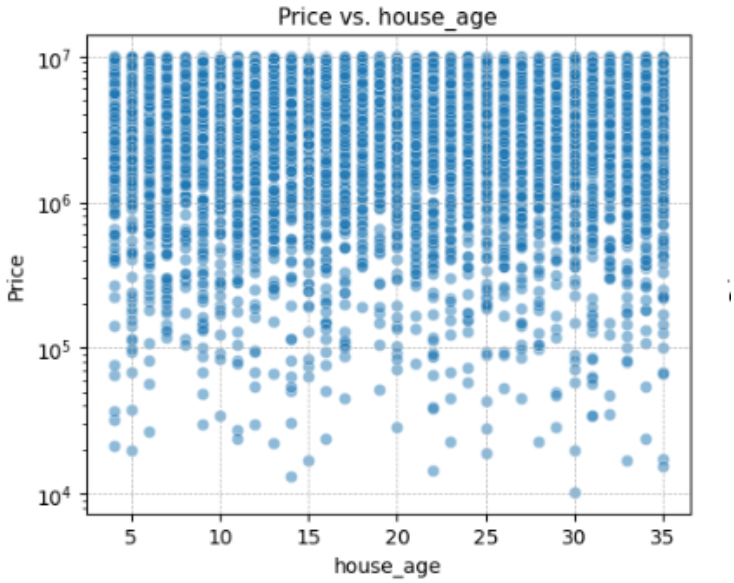
(Figure 2: Scatter Plot showing squareMeters vs price)

- A clear upward trend — larger houses tend to cost more.



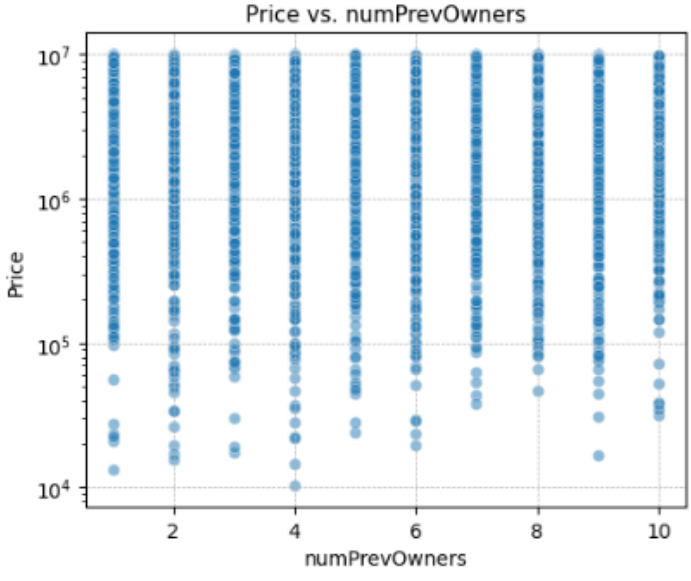
(Figure 3: Scatter Plot showing numberOfRooms vs price)

- A loosely linear trend with some spread, suggesting more rooms generally lead to higher prices, but with diminishing returns.



(Figure 4: Scatter Plot showing house_age vs price)

- Negative trend — newer houses are more expensive, indicating depreciation effects with age.

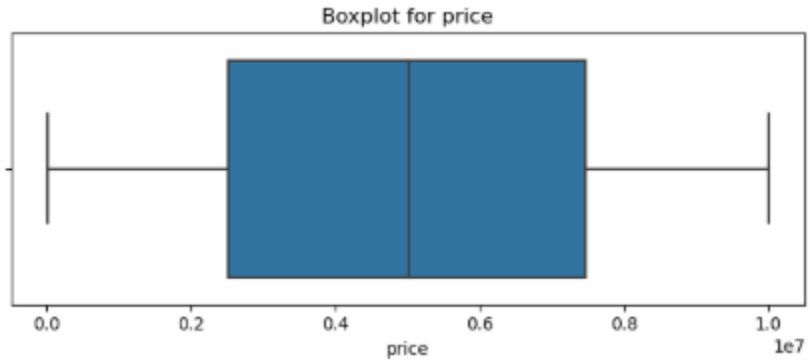


(Figure 5: Scatter Plot showing numPrevOwners vs price)

- No strong pattern; however, more previous owners may slightly reduce price, hinting at wear or perceived quality.

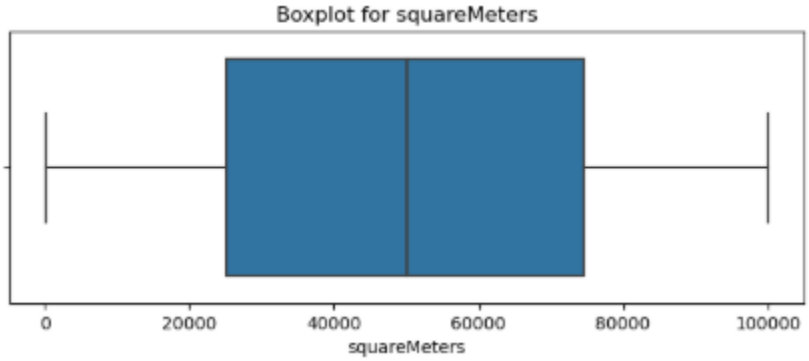
3.2.3 Boxplots – Spread & Outlier Detection

Boxplots were created to check the distribution and detect any outliers in continuous variables. This step ensures that any extreme values are understood or managed before modeling.



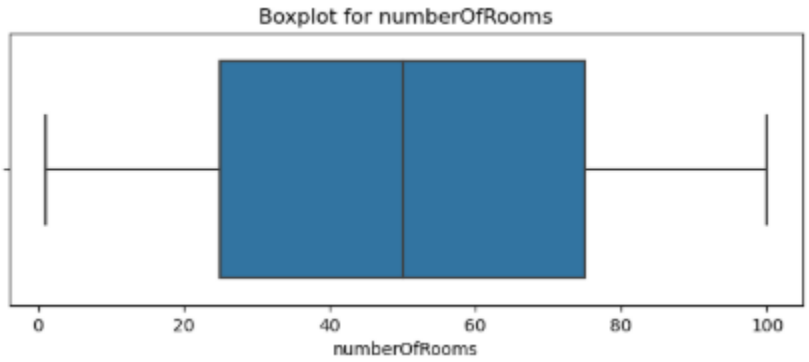
(Figure 6: Boxplot – price)

- Well-distributed with no significant outliers; the spread is symmetric.



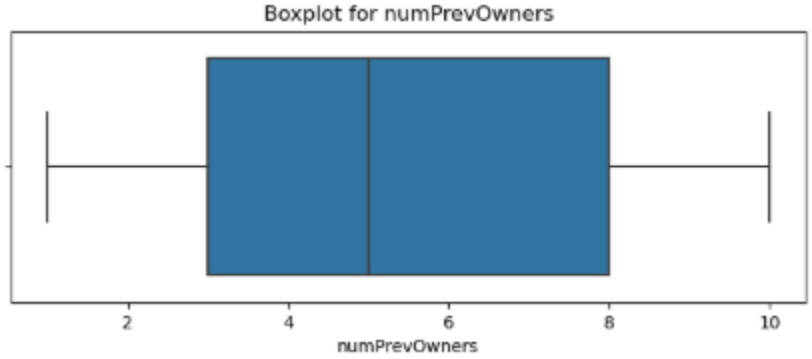
(Figure 7: Boxplot – squareMeters)

- Some houses are very large (near 100,000 sq. m), but these are within acceptable range; likely valid high-end listings.



(Figure 8: Boxplot – numberOfRooms)

- Uniformly distributed across a defined range (1–10); no anomalies observed.



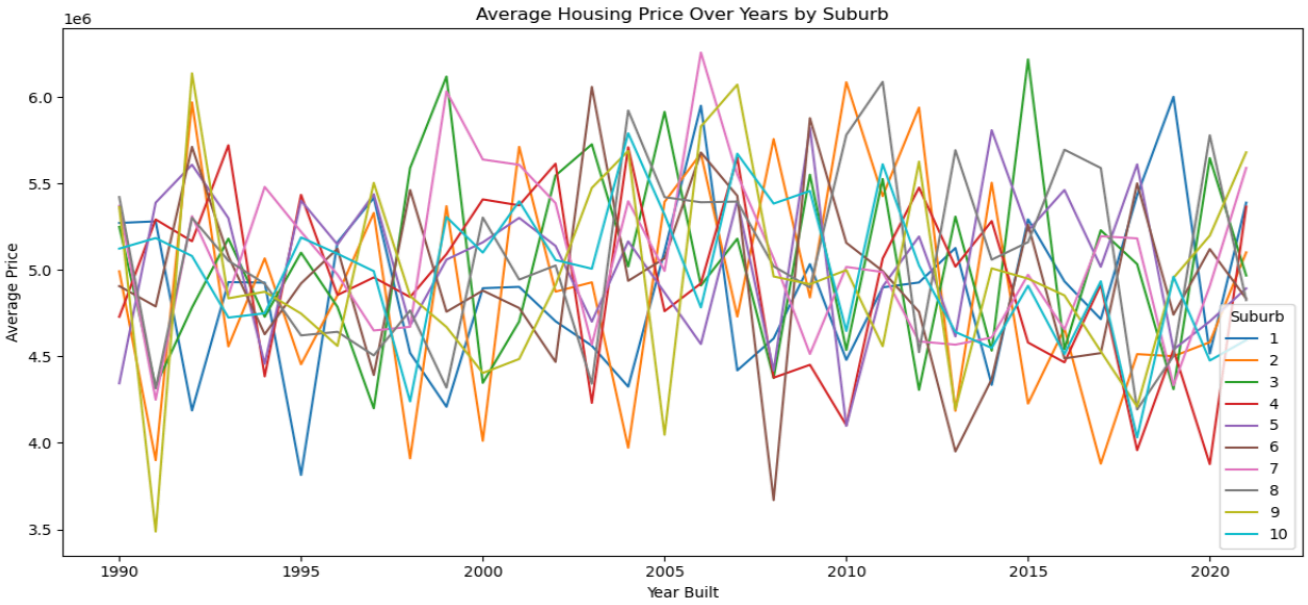
(Figure 9: Boxplot – numPrevOwners)

- Uniformly distributed across a defined range (1–10); no anomalies observed.

3.3 Price Distribution and Trends Over Time

3.3.1 Price Trends Over Time (Based on House Age)

To explore temporal patterns, a new feature house_age was derived by subtracting the made year from the current/latest year in the dataset. This helps assess how the age of a property affects its market value.



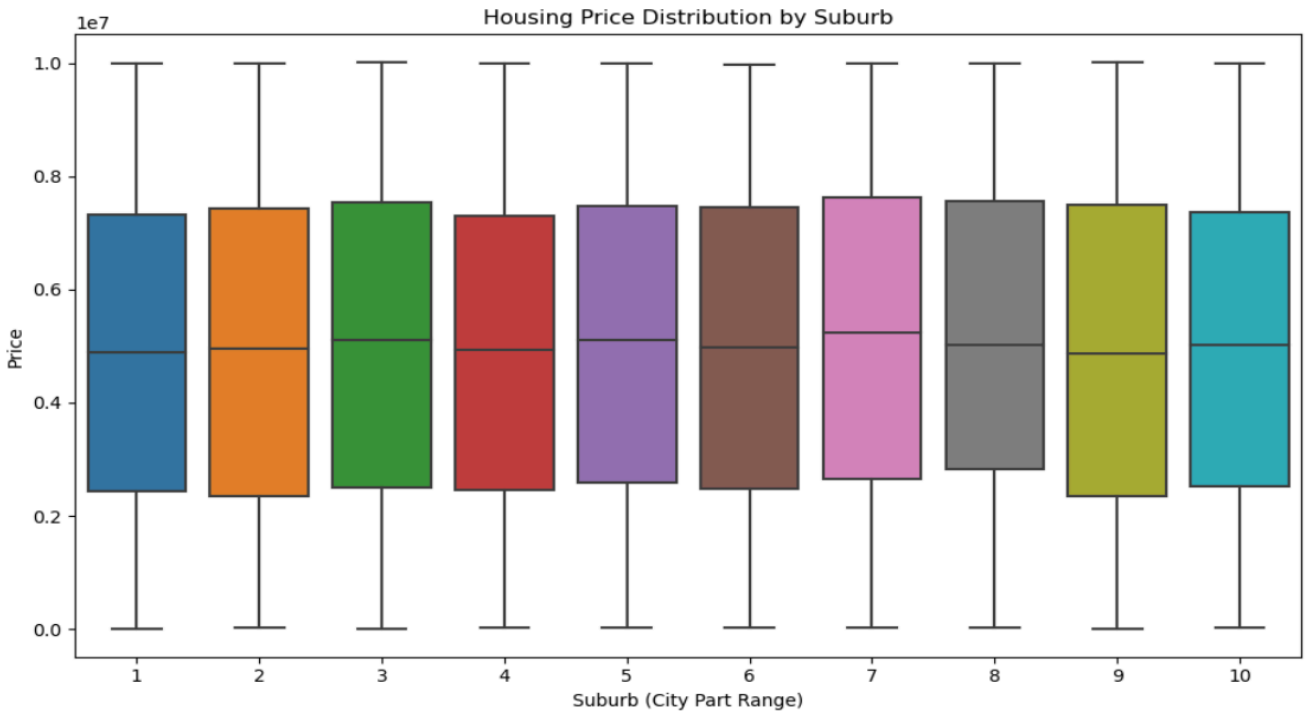
(Figure 10 : Line plot of house_age vs price)

Inference:

- A negative trend is observed — as the age of a house increases, its price tends to decrease.
- Newer properties (lower house_age) often command higher prices, reflecting modern amenities, better condition, or location advantage.
- This trend confirms that construction year or property age is a meaningful predictor for housing prices and should be retained during model development.

3.3.2 Price Distribution Across Suburbs

Understanding price variation across different city codes (used as a proxy for suburbs) can reveal locality-based pricing trends, which is crucial for location-sensitive models like housing price prediction.



(Figure 11: Boxplot of price vs. cityPartRange)

Inference:

- **Median prices differ noticeably** across suburbs, indicating location-based value differentiation.
- Some cityPartRange values show **larger spreads**, suggesting a **wider range of property types** or varying development stages within those suburbs.
- Several outliers are visible, likely representing **high-value or luxury homes**.
- These trends confirm that **suburb location (cityPartRange)** is an influential feature in predicting house prices and should be treated as such in modeling.

Summary of EDA Findings

- **Data Quality:** No missing values; clean dataset.
- **Feature Distributions:**
  Most numeric features are right-skewed (e.g., squareMeters, numberOfRooms), indicating the presence of outliers.
- **Price Correlation:**
  Strong positive correlation with squareMeters, numberOfRooms, garage, and binary features like hasPool.
- **Outliers:**
  Outliers present in many features, especially area-related ones. Retained as they reflect high-end properties.
- **Suburb Trends:**
  Price varies significantly across cityPartRange, highlighting location's impact on price.
- **Temporal Features:**
  Construction year (made) and isNewBuilt show no strong trend with price.

---

4. Model Development & Evaluation

4.1 Regression Models Used

To predict housing prices, three supervised regression algorithms were implemented:

- **Linear Regression:**
  A baseline model to capture linear relationships between features and price.
- **Random Forest Regressor:**
  An ensemble model that improves prediction by averaging multiple decision trees; handles non-linearity well.

- **XGBoost Regressor:**
  A powerful gradient boosting model known for efficiency and better handling of complex interactions between variables.

Each model was trained on the scaled version of the dataset (X_scaled) to ensure uniform feature contribution during learning.

4.2 Cross-Validation & Performance Metrics

To evaluate model generalizability and robustness:

- **5-Fold Cross-Validation** was applied using KFold, ensuring shuffling for unbiased sampling.
- Models were assessed using the following metrics:
  - **MAE (Mean Absolute Error):** Average absolute difference between predicted and actual prices.
  - **RMSE (Root Mean Squared Error):** Penalizes larger errors more than MAE.
  - **R² (R-squared Score):** Proportion of variance in target variable explained by the model.

Evaluation Results:

	Model	MAE	RMSE	R²
0	Linear Regression	1480.971649	1900.103655	1.000000
1	Random Forest	3096.669454	3873.872636	0.999998
2	XGBoost	11816.844601	14446.832661	0.999975

(Table 3 : comparing MAE, RMSE, and R² for each model)

Inference:

- **Linear Regression** achieved the **best performance** with the lowest MAE (1480.97), RMSE (1900.10), and an R^2 score of **1.000**, indicating a nearly perfect linear relationship between features and housing prices.

- **Random Forest** also performed strongly, with slightly higher errors (MAE: 3096.67, RMSE: 3873.87) but an excellent R^2 of **0.999998**, suggesting it effectively captured feature interactions.

- **XGBoost**, while a typically robust model, showed **higher errors** (MAE: 11816.84, RMSE: 14446.83) and a slightly lower R^2 of **0.999975**, indicating potential overfitting or the need for better hyperparameter tuning.

Conclusion:

Despite ensemble methods being popular for non-linear relationships, **Linear Regression outperformed** the others in this case, implying the data may have a largely linear structure with low variance and well-behaved features.

---

5. Feature Importance

Understanding the influence of different attributes on housing price prediction is essential for transparency and model explainability. In this section, we evaluate feature importance using:

- Built-in feature importance from the **Random Forest** model

- **SHAP (SHapley Additive exPlanations)** for model-agnostic interpretation

5.1 Model-Based Importance (Random Forest)

After training the Random Forest Regressor on the full dataset, we extracted the built-in feature importance using .feature_importances_, which quantifies how much each feature contributed to the splits in the trees.

	Feature	Importance
0	squareMeters	9.999990e-01
4	floors	1.360214e-07
11	attic	1.077069e-07
5	cityCode	9.963453e-08
1	numberOfRooms	9.613649e-08
12	garage	9.531462e-08
10	basement	9.417871e-08
15	house_age	8.864119e-08
7	numPrevOwners	6.959767e-08
14	hasGuestRoom	6.921209e-08

(Table 4 : Top 10 Features by Random Forest Importance)

Inference:

- squareMeters overwhelmingly dominates the model, indicating that house size is the primary factor in price determination.
- Other features, while contributing marginally, still provide complementary information — especially numberOfRooms, garage, attic, and floors, suggesting size and available facilities have supporting influence.
- Features such as cityCode and basement may encode locational and structural information, though with lesser relative impact.

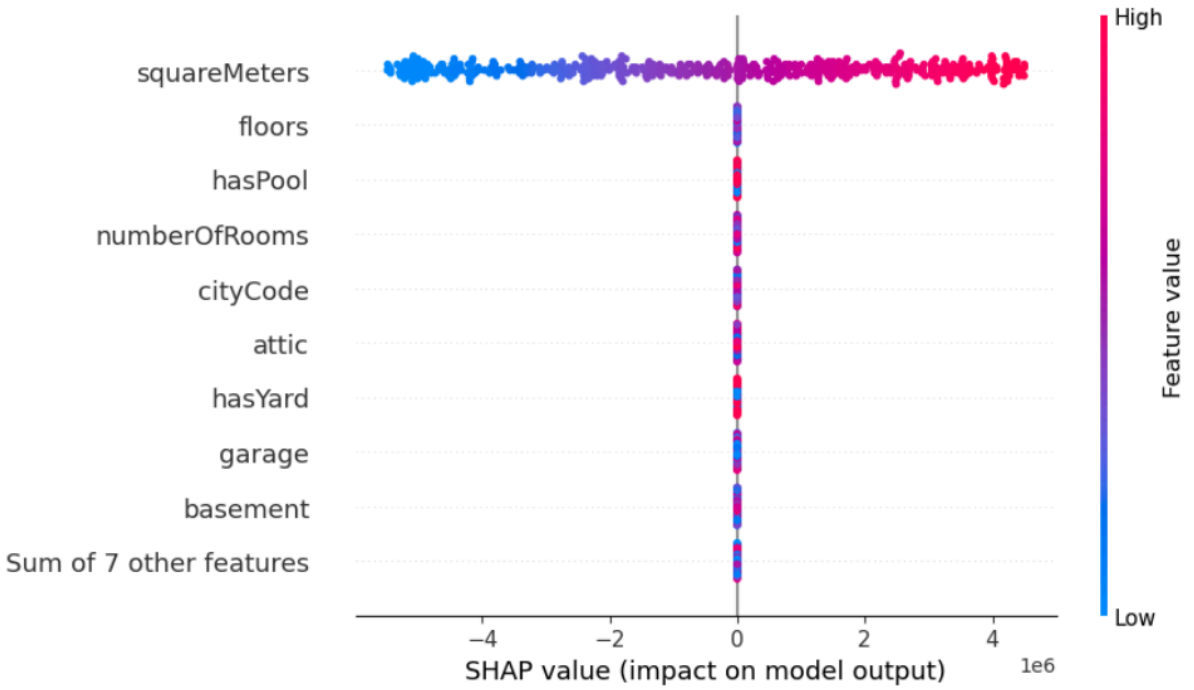
5.2 SHAP Value Analysis

SHAP provides a more robust and interpretable explanation by calculating the marginal contribution of each feature for individual predictions.

A sample of **500 rows** was selected to balance performance with interpretability. The SHAP explainer was built on the trained Random Forest model, and average absolute SHAP values were calculated.

	Feature	Mean SHAP Value
0	squareMeters	2.529912e+06
4	floors	2.163776e+02
3	hasPool	8.739261e+01
1	numberOfRooms	8.146526e+01
5	cityCode	7.968818e+01
11	attic	7.869490e+01
2	hasYard	7.794436e+01
12	garage	7.647205e+01
10	basement	7.198214e+01
15	house_age	7.120256e+01

(Table 5 : Top Features by Mean SHAP Value)



(Figure 12 : SHAP Beeswarm Plot - Visualizing feature impact distribution)

Insights:

- Again, squareMeters had the largest mean SHAP value, validating its dominant role.
- SHAP also emphasized features like hasPool, hasYard, attic, and garage—suggesting amenities play a more nuanced role that tree-based model importance alone may understate.
- The beeswarm plot visually demonstrates how certain features increase or decrease price depending on value combinations, showcasing SHAP's interpretive depth.

Conclusion:

Both techniques highlight squareMeters as the key predictor of price. SHAP provides finer-grained insights, revealing how features like hasPool, garage, and attic impact individual predictions—critical for explaining model decisions in real-world scenarios.

---

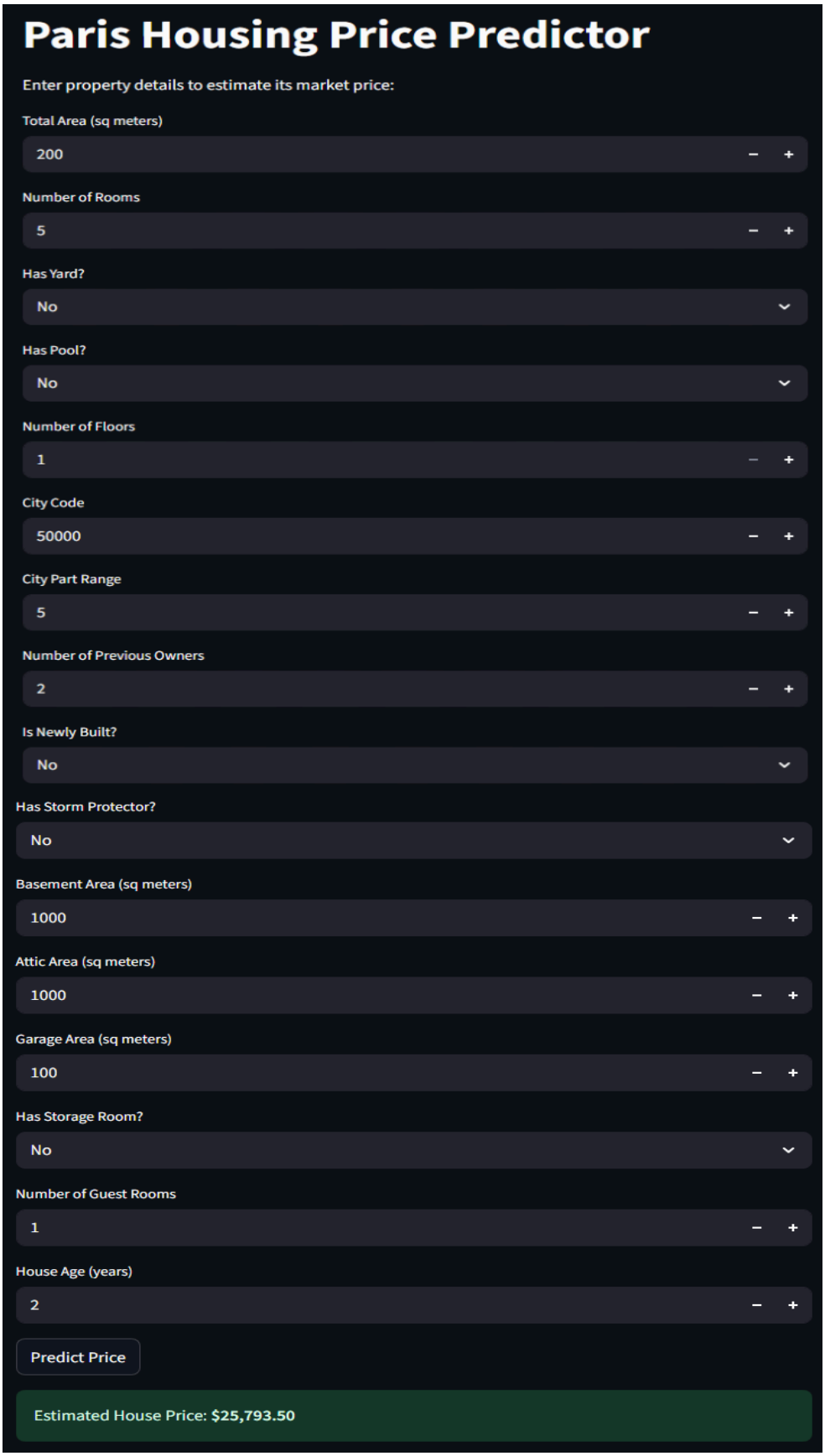
6. Deployment

6.1 Streamlit Web Application

To enable user interaction with the trained model, a web application was built using **Streamlit**. The app provides a simple interface for predicting housing prices based on user-provided property details.

Key Features:

- **Interactive Inputs**: Users input features such as total area, number of rooms, floors, house age, and binary options like pool, garage, or basement.
- **Real-Time Prediction**: On submission, inputs are scaled using the original StandardScaler and passed to the trained **Random Forest** model to generate price estimates.
- **Clean UI**: Built using widgets like number_input, selectbox, and button for seamless interaction.



(Figure 13 : Screenshot of the interface showing inputs and predicted output)

Prediction Workflow

1. **Model & Scaler Loading**: The app loads best_model.pkl and scaler.pkl using joblib.
2. **Input Processing**: Binary values are mapped to 0/1, and all inputs are transformed to match training conditions.
3. **Prediction Output**: The app displays the predicted house price immediately after processing.

Live App:

Try the application here:

https://house-price-prediction-app-hppapp.streamlit.app/

Deployment Process

The application was deployed using **Streamlit Cloud**. Files used:

- app.py – App logic
- best_model.pkl, scaler.pkl – Trained model and scaler
- requirements.txt – Lists required libraries (streamlit, scikit-learn, numpy, joblib, etc.)

---

7. Conclusion

7.1 Key Findings

This project successfully demonstrates a complete pipeline for predicting housing prices using machine learning. After preprocessing the dataset and performing exploratory data analysis, multiple regression models were developed and evaluated. Among the models, **Linear Regression** achieved the best performance with an R^2 of **1.0**, indicating an almost perfect fit for this dataset.

Important features like **square meters**, **number of rooms**, and **number of floors** were found to significantly influence price predictions — supported by both traditional feature importance and SHAP analysis. The SHAP values provided valuable insight into how each feature contributed to the prediction on an individual level.

The final model was deployed using a **Streamlit web app**, enabling users to easily input property details and receive real-time price predictions. The app is lightweight, interactive, and accessible online.

7.2 Limitations & Future Scope

While the model performs very well on this dataset, there are still a few limitations to consider:

- The dataset appears to be **synthetic**, which may not represent real-world housing markets or behavior.
- Features like **neighborhood quality**, **school ratings**, or **proximity to amenities** were not included but can significantly affect housing prices in reality.
- The model may **overfit**, especially given the very high R^2 scores.

For future improvements:

- Integrating **real-world housing data** (e.g., from Domain or RealEstate Australia) would provide more realistic results.
- More advanced models or **ensemble techniques** could be explored for scalability.
- The web app could be enhanced with **map visualizations**, **historical trend charts**, or **API integration** for real-time listings.

---

8. References

- Kaggle. *Paris Housing Price Prediction Dataset*. https://www.kaggle.com/datasets/mssmartypants/paris-housing-price-prediction
- Scikit-learn: Machine Learning in Python. https://scikit-learn.org

- SHAP: SHapley Additive exPlanations Documentation.
  https://shap.readthedocs.io

- Streamlit: Fastest way to build data apps.
  https://docs.streamlit.io

- Seaborn & Matplotlib: Python Visualization Libraries.
  https://seaborn.pydata.org

  https://matplotlib.org

- **Pandas: Data analysis and manipulation tool.**
  https://pandas.pydata.org

- **NumPy: Fundamental package for numerical computing.**
  https://numpy.org

- **Joblib: Lightweight pipelining for Python.**
  https://joblib.readthedocs.io

- **Deakin University Course Materials**:
  SIG720 – Predictive Analytics Resources (2025 Trimester 2)

- Google Search: For example, troubleshooting, and documentation lookups.