Project Report: House Price Prediction

1. Executive Summary

This project aims to build a predictive model to estimate house prices based on various property features such as area, number of bedrooms, location, and year built. Following a structured data science lifecycle, we performed data exploration, preprocessing, and trained multiple regression models. The initial results from Linear Regression, Decision Tree, and Random Forest models were unsatisfactory, with all models performing worse than a baseline model that predicts the mean house price. This report details the process, findings, and provides recommendations for future improvements to achieve a viable predictive solution.

---

2. Project Phases & Methodology

Phase 1: Data Understanding & Exploration

Dataset Source: House Prediction Dataset from Kaggle **Target Variable:** `Price`

Data Description: The dataset contains **2,000 records** and **10 columns** with a mix of numerical and categorical features.

Column Name	Data Type	Description
Id	int64	Unique identifier for each property
Area	int64	Area of the property (sq. ft.)
Bedrooms	int64	Number of bedrooms
Bathrooms	int64	Number of bathrooms
Floors	int64	Number of floors
YearBuilt	int64	Year the property was built
Location	object	Categorical (e.g., Downtown, Suburban)
Condition	object	Categorical (e.g., Excellent, Good, Fair)
Garage	object	Categorical (Yes/No)
Price	int64	Target variable: Sale price of the house

Phase 2: Data Preprocessing

The following preprocessing steps were applied:

1. **Feature-Target Split:** The `Price` column was separated as the target variable `y`.
2. **Encoding Categorical Variables:** The categorical features (`Location`, `Condition`, `Garage`) were converted into numerical format using `LabelEncoder`.
3. **Feature Scaling:** All numerical features were standardized using `StandardScaler` to ensure each feature contributed equally to the model.

4. **Train-Test Split:** The processed data was split into training (80%) and testing (20%) sets.

---

3. Data Quality Report

A preliminary analysis was conducted to assess the quality and characteristics of the dataset.

Summary Statistics:

Statistic	Id	Area	Bedrooms	Bathrooms	Floors	YearBuilt	Price
Count	2000.0	2000.0	2000.0	2000.0	2000.0	2000.0	2000.0
Mean	1000.5	2786.2	3.0	2.55	1.99	1961.4	537,676.9
Std	577.5	1295.1	1.42	1.11	0.81	35.93	276,428.8
Min	1.0	501.0	1.0	1.0	1.0	1900.0	50,005.0
25%	500.8	1653.0	2.0	2.0	1.0	1930.0	300,098.0
50%	1000.5	2833.0	3.0	3.0	2.0	1961.0	539,254.0
75%	1500.2	3887.5	4.0	4.0	3.0	1993.0	780,086.0
Max	2000.0	4999.0	5.0	4.0	3.0	2023.0	999,656.0

Key Observations from Data Quality:

- **Completeness:** The dataset has no missing values (all columns show 2000 non-null entries), which is excellent.
- **Data Types:** All data types are appropriate (`int64` for numerical, `object` for categorical).
- **Outliers:** A box plot analysis was performed on all numerical features. The analysis concluded that **the dataset does not contain any significant outliers**. The values for `Area`, `YearBuilt`, and `Price` are within a reasonable and expected range for a housing dataset.
- **Inconsistencies:** No obvious inconsistencies were reported. The categorical variables (`Location`, `Condition`, `Garage`) have values that align with their expected definitions.

---

4. Model Training & Evaluation

Three different regression models were trained and evaluated on the test set.

Evaluation Metrics:

- **Mean Squared Error (MSE):** Measures the average squared difference between actual and predicted values. Lower is better.
- **R-squared (R^2):** Represents the proportion of variance in the target variable explained by the model. Closer to 1 is better. A **negative R^2** indicates the model is worse than simply predicting the mean of the target.

Results Summary

Model	MSE (Billions)	R^2 Score	Cross-Val R^2 (Mean)	Key Observations
Linear Regression	78.66	-0.011	-0.005	Best of the three, but still poor. Fails to capture non-linear patterns.
Decision Tree	167.22	-1.149	N/A	Worst performer. Severe overfitting; predictions are highly erratic.
Random Forest	82.68	-0.063	N/A	Better than a single tree but still worse than Linear Regression.

Visual Comparison: A plot of the first 100 samples from the test set clearly shows the models' performance:

- The **Decision Tree** predictions are wildly inconsistent and do not follow the actual price trend.
- The **Random Forest** predictions are smoother but still largely inaccurate.
- The **Linear Regression** predictions form a smooth line that fails to capture the fluctuations in actual prices, though it is the closest to the general range.

 Actual vs Predicted Prices **(Note: This is a description of the plot you generated. In a real report, you would insert the image here.)***

---

5. Conclusion & Recommendations

Conclusion

The initial modeling effort was unsuccessful. All three models produced negative R^2 scores, meaning they are less accurate than a simple baseline model that always predicts the average house price. The Linear Regression model was the least poor, followed by Random Forest and then the severely overfit Decision Tree.

The core issue is that the initial features and model configurations are insufficient to capture the complex, non-linear relationships that determine house prices in this dataset.

Recommendations for Future Work

1. **Advanced Feature Engineering:**
    - Create new features like `AgeOfHouse` (Current Year - `YearBuilt`).
    - Investigate interaction terms (e.g., `Area` per `Bedroom`).
    - Apply more nuanced encoding for ordinal categories like `Condition` (e.g., Fair=0, Good=1, Excellent=2) instead of `LabelEncoder`.
2. **Algorithm Exploration:**
    - **Gradient Boosting Machines (XGBoost, LightGBM, CatBoost):** These are powerful algorithms designed for tabular data and often outperform Random Forests.
    - **Support Vector Machines (SVR):** With appropriate kernel functions, SVR can model non-linear relationships.
    - **Neural Networks:** A well-designed neural network can capture complex interactions between features.

3. **Hyperparameter Tuning:** Perform a grid or random search to find the optimal parameters for the Random Forest and other models. The current default parameters are clearly not optimal.

4. **Data Collection:** If possible, gather more relevant data such as proximity to amenities, school district ratings, or recent renovation history, which could be strong price predictors.

5. **Further Data Investigation:** Re-examine the assumption of "no outliers." While the box plots might not show extreme points, a model's poor performance can sometimes be caused by influential points that require closer statistical inspection.

$$ \boxed{ \text{Final Summary: Linear Regression > Random Forest > Decision Tree (in terms of performance). However, all models are underperforming and require significant further improvements.} } $$
