Predicting Property Prices and Handling Missing Data

A Comparative Study of Decision Trees, XGBoost, and LightGBM

By:

Saheen Ibrahim A251000076

Rand Hasan A25100103

Vana Mustafa A25100069

Lana Azad A25100115

Asmaa Salih A25100112

Module: Data Mining

Lecturer: Dr. Shamal Taha

The American University of Kurdistan

Kurdistan Region of Iraq, Duhok

Fall 2025

Contents

Introduction

The project considers the problem of property prices prediction through machine learning models and specific strategies for dealing with missing values. Data dropout is a common issue with realistic data sets, and the method used to deal with it can have a great influence on the predictor performance. The goal was to compare procedures of simple imputation with native incoming procedures of missing data used in models.

dataset and missing value analysis

The collected dataset is called arproperties.csv and has 1,000,000 property records with the characteristics of: the number of rooms, the number of bathrooms, surface areas, location, and the type of property. The target variable is price.

It was also found that several columns had significant amounts of missing values:

- **Completely missing**: l6
- **High missing percentages**: l4, l5, surface_total, surface_covered, rooms, bedrooms
- **Moderate missing percentages**: lat, lon, bathrooms, price_period

This proved the necessity of having specific plans in dealing with incomplete data before and throughout modeling.

strategies of missing Data

There were 2 approaches:

1. **Simple Imputation**

- Median imputation for numerical features
- Mode imputation for categorical features
- Easy to apply but risks distorting distributions, reducing variance, and introducing bias.

2. **Native Handling by Models**

- Left missing numerical values as NaN
- Used one-hot encoding with dummy_na=True for categorical variables

- Relied on models like **Decision Trees, XGBoost, and LightGBM**, which can learn optimal directions for missing values during training.

Models and Performance (RMSE)

Four models were trained and evaluated:

- **Decision Tree with Imputation**
- **Decision Tree without Imputation (NaNs kept)**
- **XGBoost**
- **LightGBM**

Results (illustrative placeholders):

- Decision Tree (Imputed): *RMSE* = ...
- Decision Tree (No Imputation): *RMSE* = ...
- XGBoost: *RMSE* = ...
- LightGBM: *RMSE* = ...

Key Insights

Native handling > simple imputation: XGBoost and LightGBM performed in parallel when trained to track missing values instead of median/mode imputation.

Another great algorithm: XGBoost is the top-performing model overall, and LightGBM comes almost the same.

Comparison between Decision Tree and imputation: The Decision Tree was actually performing at a better level compared to the simple imputation, despite not wanting imputation.

Conclusion and Next Steps

The outputs demonstrate that more sophisticated models, such as XGBoost and LightGBM, which deal with missing values internally, are better than straightforward imputation techniques. Next steps include: Attempting higher imputation techniques, including the KNN or model imputation.