Purwadhika
Digital Technology School

APARTMENT PRICE PREDICTION MODEL

Sulaeman Nurhakim
JCDS 2804-001

May 14th 2025

Business Problem Understanding

Background

- Daegu is experiencing rapid urban growth.
- Increased demand for apartments leads to price fluctuations.
- Pricing is influenced by multiple factors (location, size, age, etc).
- Objective: Use machine learning to predict apartment sale prices based on historical data.

Problem Statements

- How can we use apartment data to predict sale prices?
- Which machine learning model performs best?
- What are the key factors that influence apartment prices?

Goals

- Build several machine learning models to predict apartment prices.
- Evaluate model performance using standard metrics.
- Identify which features most affect the final sale price.

Methodology

Step-by-step workflow:

1. Data Understanding – Explore dataset features and structure.
2. Preprocessing – Handle missing values, encode categorical data, scale numerical values.
3. EDA (Exploratory Data Analysis) – Visualize trends, patterns, and correlations.
4. Modeling – Apply multiple regression models.
5. Evaluation – Compare models using MAE, RMSE, and R^2 score.
6. Interpretation – Identify key factors influencing apartment prices.

Metrics Evaluation

- The evaluation metrics used for the regression model are RMSE, MAE, and MAPE.
- RMSE is the root of the mean square residual.
- MAE is the average of absolute errors and is a metric that is not sensitive to outliers.
- MAPE is the absolute average percentage error.

The smaller the RMSE, MAE, and MAPE values, **the better** the model will be at predicting apartment prices.	R-Squared can also be used if the best model chosen is a linear model. **The closer** the value is to **1**, **the better** the regression line will represent the data.

Dataset Overview

	HallwayType	TimeToSubway	SubwayStation	N_FacilitiesNearBy(ETC)	N_FacilitiesNearBy(PublicOffice)	N_SchoolNearBy(University)	N_Parkinglot(Basement)	YearBuilt	N_FacilitiesInApt	Size(sqf)	SalePrice
0	terraced	0-5min	Kyungbuk_uni_hospital	0.0	3.0	2.0	1270.0	2007	10	1387	346017
1	terraced	10min~15min	Kyungbuk_uni_hospital	1.0	5.0	1.0	0.0	1986	4	914	150442
2	mixed	15min~20min	Chil-sung-market	1.0	7.0	3.0	56.0	1997	5	558	61946
3	mixed	5min~10min	Bangoge	5.0	5.0	4.0	798.0	2005	7	914	165486
4	terraced	0-5min	Sin-nam	0.0	1.0	2.0	536.0	2006	5	1743	311504

Feature

- **Hallway Type** – Type of apartment based on hallway structure
- **TimeToSubway** – Time to reach nearest subway station
- **SubwayStation** – Name of nearest subway station
- **Nearby Facilities (ETC)** – Number of other miscellaneous facilities nearby
- **Public Offices Nearby** – Number of public office facilities nearby
- **Universities Nearby** – Number of nearby universities
- **Basement Parking Lots** – Number of basement parking spaces
- **Year Built** – Year the apartment was constructed
- **Facilities Inside Apartment** – Number of internal apartment facilities
- **Size (sqft)** – Apartment size in square feet
- **Sale Price** – Apartment selling price (in Korean Won)

Exploratory Data Analysis

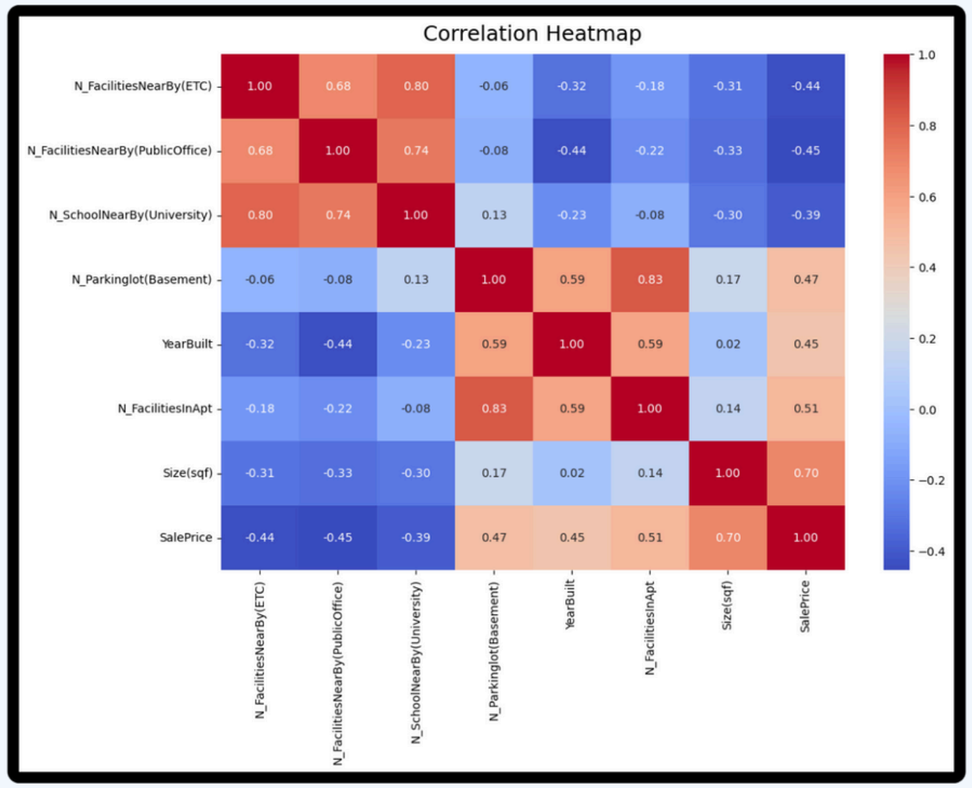
- Summary Stats

	HallwayType	TimeToSubway	SubwayStation
count	4123	4123	4123
unique	3	5	8
top	terraced	0-5min	Kyungbuk_uni_hospital
freq	2528	1953	1152

- **Dataset:** 4,123 rows, 11 features
- **Most Common Type:** Terraced apartment
- **Station Access:**
  - Most are within 0–5 minutes walking distance
  - Nearest station: Kyungbuk Uni Hospital Station
- **Nearby Facilities:**
  - Avg: 2 facilities, 4 public offices, 3 universities, 569 basement parking spots
- **Year Built:**
  - Avg: 2003 (Oldest: 1978, Newest: 2015)
- **Internal Facilities:**
  - Avg: 6, Max: 10
- **Size & Price:**
  - Avg size: 954.63 sq ft
  - Avg price: 221,767.93 won
  - Avg price per sq ft: 232.31 won

	N_FacilitiesNearBy(ETC)	N_FacilitiesNearBy(PublicOffice)	N_SchoolNearBy(University)	N_Parkinglot(Basement)	YearBuilt	N_FacilitiesInApt	Size(sqf)	SalePrice
count	4123.000000	4123.000000	4123.000000	4123.000000	4123.000000	4123.000000	4123.000000	4123.000000
mean	1.930876	4.135338	2.746301	568.979141	2002.999757	5.817851	954.630851	221767.926995
std	2.198832	1.802640	1.496610	410.372742	8.905768	2.340507	383.805648	106739.839945
min	0.000000	0.000000	0.000000	0.000000	1978.000000	1.000000	135.000000	32743.000000
25%	0.000000	3.000000	2.000000	184.000000	1993.000000	4.000000	644.000000	144752.000000
50%	1.000000	5.000000	2.000000	536.000000	2006.000000	5.000000	910.000000	209734.000000
75%	5.000000	5.000000	4.000000	798.000000	2008.000000	7.000000	1149.000000	291150.000000
max	5.000000	7.000000	5.000000	1321.000000	2015.000000	10.000000	2337.000000	585840.000000

Data Correlation

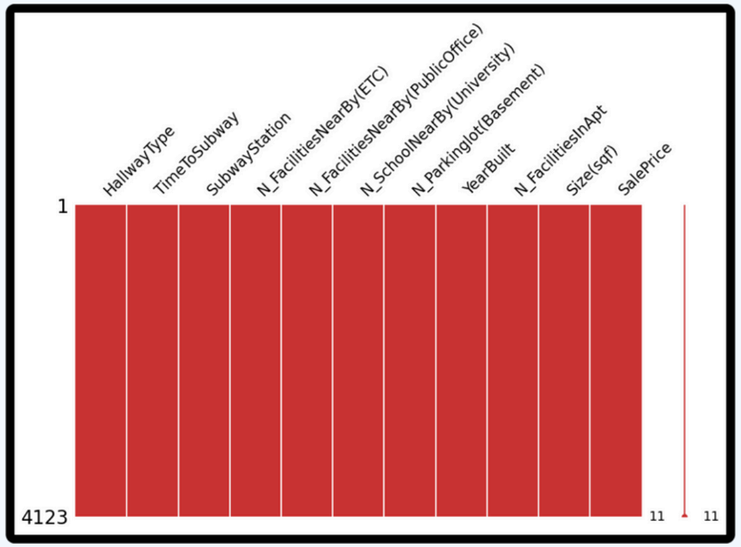


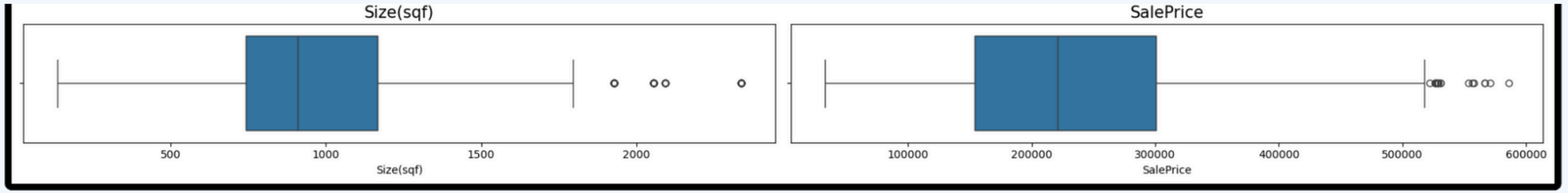
- **Medium Correlation with Price**
  - All features show a medium correlation (0.3–0.7) with apartment prices.
  - This indicates that each feature has a meaningful relationship with the sale price.
- **Strong Correlation Between Independent Variables**
  - Number of nearby facilities ↔ Number of nearby universities: correlation 0.80
  - Public offices nearby ↔ Number of nearby universities: correlation 0.74
  - This indicates the presence of multicollinearity.
- **What is Multicollinearity?**
  - When two or more independent variables are highly correlated with each other. It can cause:
    - Difficulty in interpreting regression coefficients
    - Unstable parameter estimates
    - Reduced model accuracy
- **Impact on Regression Models**
  - Especially affects models like: Linear Regression, Logistic Regression
  - Recommended actions:
    - Drop or combine highly correlated variables, Use regularization techniques (e.g., Ridge, Lasso)

Data preprocessing

In general it can be seen that:

- There are 4,123 data and 11 columns in the dataset.
- There are 3 categorical columns and 8 numerical columns.
- There is no empty data in all dataset columns.
- Several columns in the data have a value of 0 which can be interpreted as the absence of facilities (public offices, universities, basement parking lots, etc.) around the apartment.
- There are 1,422 duplicate records in the dataset, making up 34.49% of the data.
- Duplicate data can lead to model bias and overfitting, as the same data points are counted multiple times. Therefore, we will remove all duplicates. After removing the duplicates, 2,701 records remain from the original 4,123.





Column	Outliers
Size(sqft)	84 data
SalePrice	17 data

Modelling

Feature Engineering

- Scalling

Feature	Scaling Method	Reason
`N_Parkinglot(Basement)`	Robust Scaler	To handle outliers and skewed distribution, improving data consistency and reducing extreme value influence.
`Size(Sqf)`	Robust Scaler	To handle outliers and skewed distribution, improving data consistency and reducing extreme value influence.

- Encoding

Feature	Encoding Method	Reason
`HallwayType`	One-Hot Encoding	A nominal variable with 3 categories; One-Hot Encoding is suitable due to the small number of categories.
`SubwayStation`	Binary Encoding	A nominal variable with 8 categories; Binary Encoding is used to reduce the number of dummy variables and minimize overfitting.
`TimeToSubway`	Ordinal Encoding	An ordinal variable with categories ordered based on the time to reach the nearest station.

X / Features :

- HallwayType
- TimeToSubway
- SubwayStation
- N_FacilitiesNearBy(ETC)
- N_FacilitiesNearBy(PublicOffice)
- N_SchoolNearBy(University)
- N_Parkinglot(Basement), YearBuilt
- N_FacilitiesInApt, Size(sqf)

Y / Target :

- Sales Price

Train Test Distribution :

- 80 : 20

Modelling

Benchmark Model

- Linear Regression
- Ridge Regression
- Lasso Regression
- Support Vector Regressor
- KNN Regressor
- Decision Tree
- Random Forest
- XGBoost Regressor

Best Model By Cross-Validation

Model	Mean R2	Standar Deviasi R2	Mean RMSE	Standard Deviasi RMSE	Mean MAE	Standard Deviasi MAE	Mean MAPE	Standard Deviasi MAPE
Decision Tree Regressor	0.807112	0.013265	-46270.785489	884.248170	-37259.323556	955.537365	-0.190018	0.001301
Random Forest Regressor	0.807865	0.013370	-46178.994331	885.653165	-37214.822617	965.821287	-0.190385	0.001295
XGBoost Regressor	0.806841	0.013167	-46303.876563	865.293541	-37297.524219	922.828733	-0.190406	0.001068
Voting Regressor	0.801248	0.010692	-46984.605940	550.603985	-38382.128182	710.306153	-0.197593	0.003136
KNN Regressor	0.782673	0.017688	-49095.720081	1221.113625	-39484.083333	1473.590874	-0.202705	0.004851
Stacking Regressor	0.781767	0.024490	-49143.600000	1739.926253	-39401.187500	1778.271800	-0.203574	0.006333
Linear Regression	0.755169	0.010342	-52161.731578	561.053598	-42090.448273	748.236544	-0.216895	0.005181
Lasso Regression	0.755168	0.010343	-52161.801519	561.102159	-42090.970775	748.139394	-0.216901	0.005173
Ridge Regression	0.755153	0.010346	-52163.415479	562.701170	-42099.964056	747.786206	-0.217047	0.005091
SVR	-0.007083	0.005022	-105855.515131	2178.696754	-85494.536366	3745.076433	-0.559577	0.049023

Prediction From Testing Data With Benchmark 3 Best Models

	R2	RMSE	MAE	MAPE
XGBoost	0.783239	47985.301750	38951.312500	0.198598
Random Forest	0.781391	48189.460034	39010.786937	0.200358
Decision Tree	0.777481	48618.452969	39169.162646	0.202106

Hyperparameter Tuning

XGBoost Hyperparameter Search Space

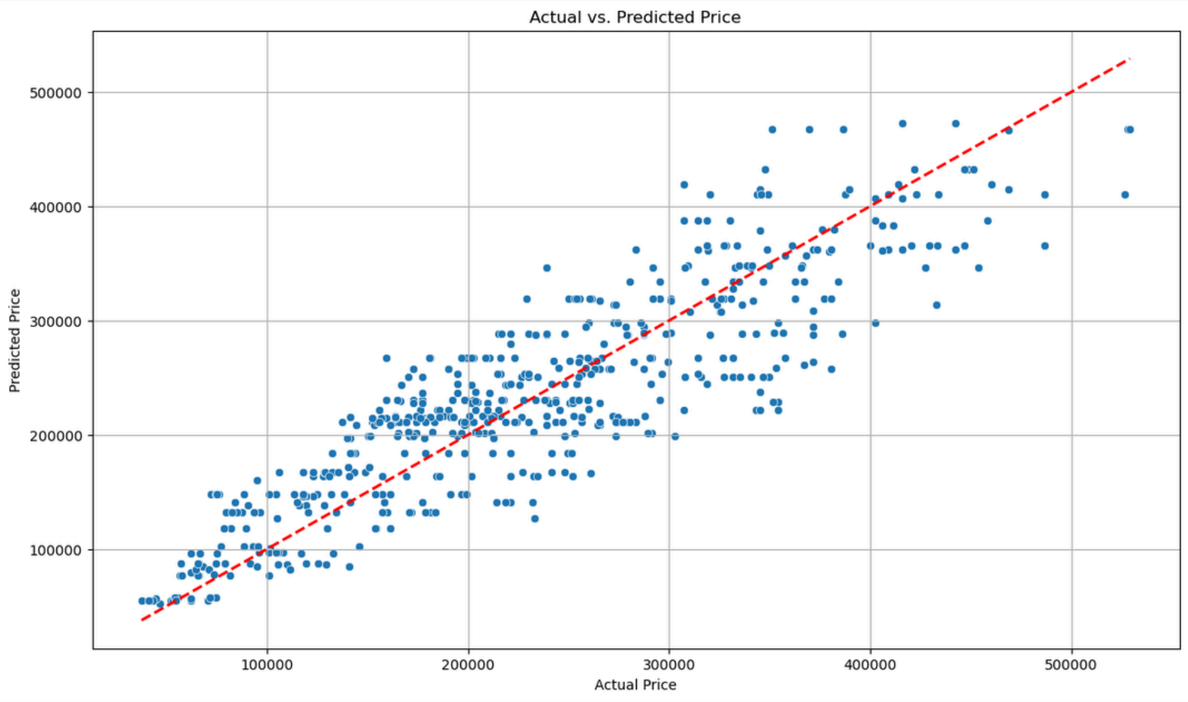
Hyperparameter	Description	Range
max_depth	The maximum depth of each tree in the XGBoost model.	1 to 10
learning_rate	The learning rate used by the XGBoost model.	0.01 to 1
n_estimators	The number of trees in the XGBoost model.	100 to 200
subsample	The percentage of the training data used for each tree in the XGBoost model.	20% to 90%
gamma	The minimum reduction in impurity required to split a leaf node in the XGBoost model.	1 to 10
colsample_bytree	The percentage of features used for each tree in the XGBoost model.	10% to 90%
reg_alpha	The regularization alpha used in the XGBoost model.	0.001 to 10

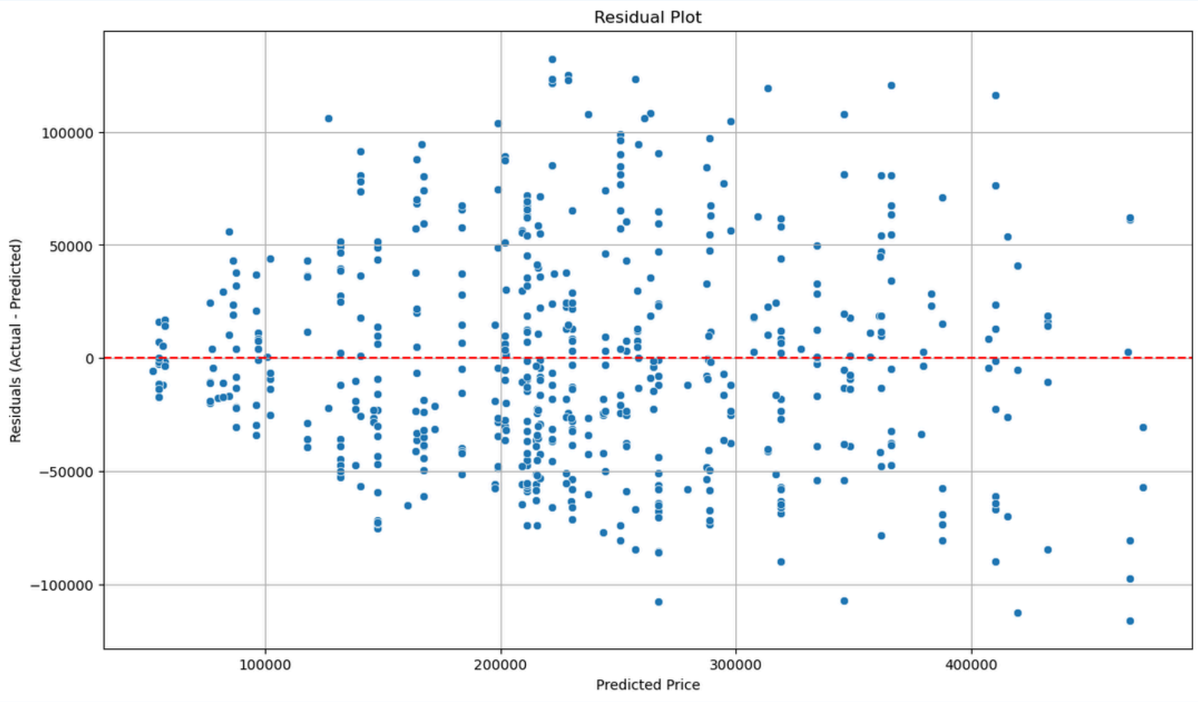
XGBoost	R2	RMSE	MAE	MAPE
Before Tuning	0,783239	47985.30175	38951.3125	0.198598
After Tuning	0,78241	48076.943663	38658.09375	0.194595

Best Hyperparameter for XGBoost :

- subsample : 0.8
- reg_alpha : 3.593813663804626
- n_estimators : 154
- max_depth: 3
- learning_rate: 0.77
- gamma : 4
- colsample_bytree: 0.9

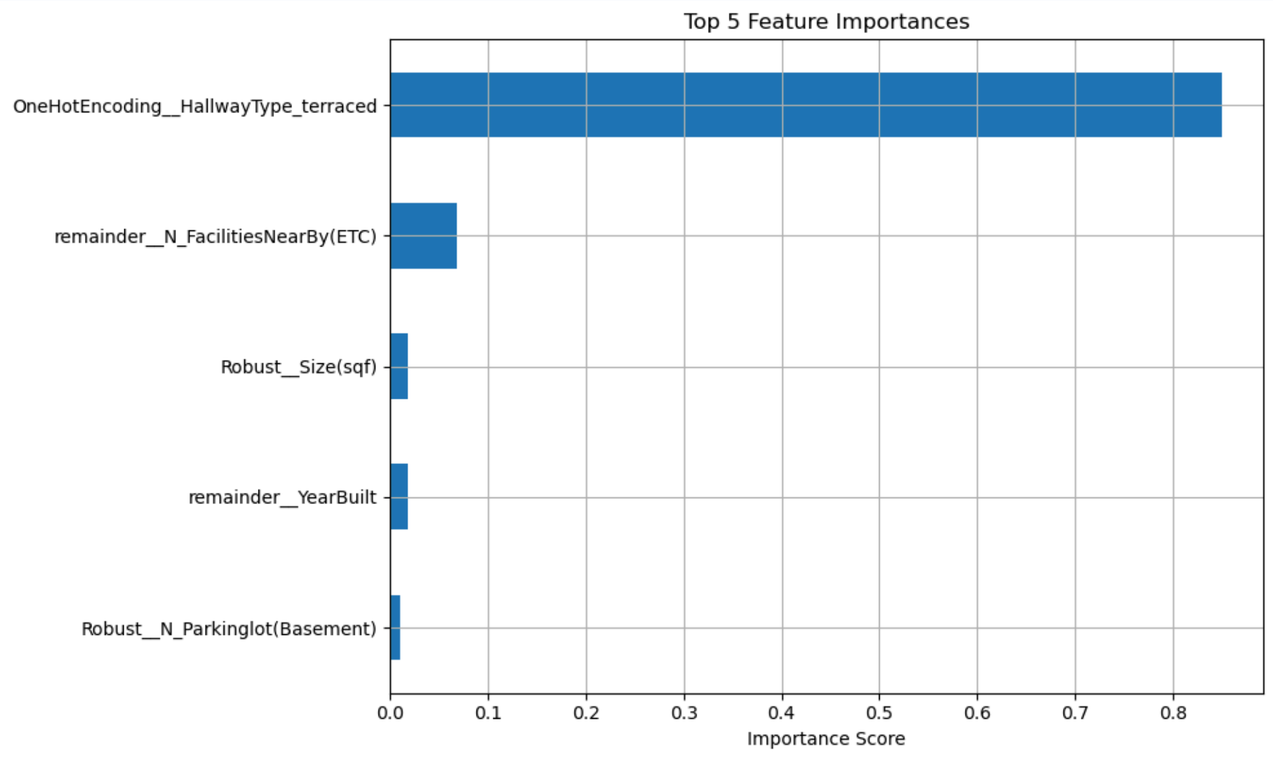
Residual Analysis





Based on the residual plot above, the residuals appear to be randomly distributed along the horizontal axis and do not have a particular pattern, indicating that the regression model is generally suitable for apartment price data in Daegu, South Korea.

Features Importances



Type of Terraced Apartment

- The type of apartment (e.g., terraced) significantly affects the price, as it often reflects building quality and living standards.

Number of Nearby Facilities

- Apartments located near more facilities (like schools, stations, hospitals) tend to have higher prices due to better accessibility and convenience.

Size of the Apartment

- Larger apartments generally have higher prices. Size is one of the most direct indicators of value.

Conclusion

Model Summary

- Final model: XGBoost Regressor
- Main metric: MAPE = 19.4%
- Interpretation: Good prediction accuracy (Lewis, 1982)

Prediction Meaning

- Price range: ₩32,743 – ₩521,902
- Prediction error: On average, 19.4% off from actual price

Key Features

- Apartment type (terraced)
- Nearby facilities
- Apartment size

Limitations

- Limited features may cause bias
- Some factors affecting price not captured

Business Usefulness

- Helps set accurate prices
- Understand what drives apartment value
- Predict price changes with feature changes

Impact

- From raw data → to actionable insight
- Real estate agents can make better pricing decisions

Recomendation

1. Add More Relevant Features

To better predict apartment prices, include:

- Floor level of the apartment
- Year of sale
- Number of rooms (bedrooms, bathrooms, kitchen)
- Furniture inclusion (fully furnished, semi, or unfurnished)
- Other apartment-specific amenities

2. Update the Dataset

- Collect newer and more recent data
- Ensure the data reflects current market trends
- Improves relevance and accuracy of the model

3. Benefits of Improvement

- More accurate and robust regression model
- Better understanding of price-driving factors
- More reliable predictions for real-world use

Thank You!