DTI

Purwadhika

Apartment Price Prediction

Daegu Apartment, Korea

Created by:
Naufal Daffa Abdurahman

ndaffaabdurahman@gmail.com

Portfolio

LinkedIn

DTI

Table Of Content

During work for this project, the key steps are outlined beside:

01 Business Understanding

02 Data Understanding

03 EDA

04 Data Preprocessing

05 Modelling

06 Interpretation

07 Conclusion

08 Recommendation

09 Limitation

DTI

Context

- Daegu is South Korea's **third-largest urban** agglomeration after Seoul and Busan.
- The third-largest official metropolitan area in the country with over **2.5 million residents**
- The **second-largest city** in the Yeongnam region in the southeastern Korean Peninsula after Busan.

Problem Statement

As of December 2023, the number of unsold apartments nationwide reached **68,000 units**, the highest in seven years since 2015.

This **increase in unsold units** is more pronounced outside Seoul, where the unsold inventory rose by **19.8%.**

01 **Oversupply**

02 **Financial Challenge**

03 **Market Instability**

Goal of this Project

As startup focused on real estate technology consultant, This model aims to:

Estimate Price

Provide realistic price estimates for apartment units.

Affordability

Ensure housing affordability for potential buyers.

Competitive Price

Support property owners in setting competitive prices

Stabilize Market

Help stabilize the local real estate market

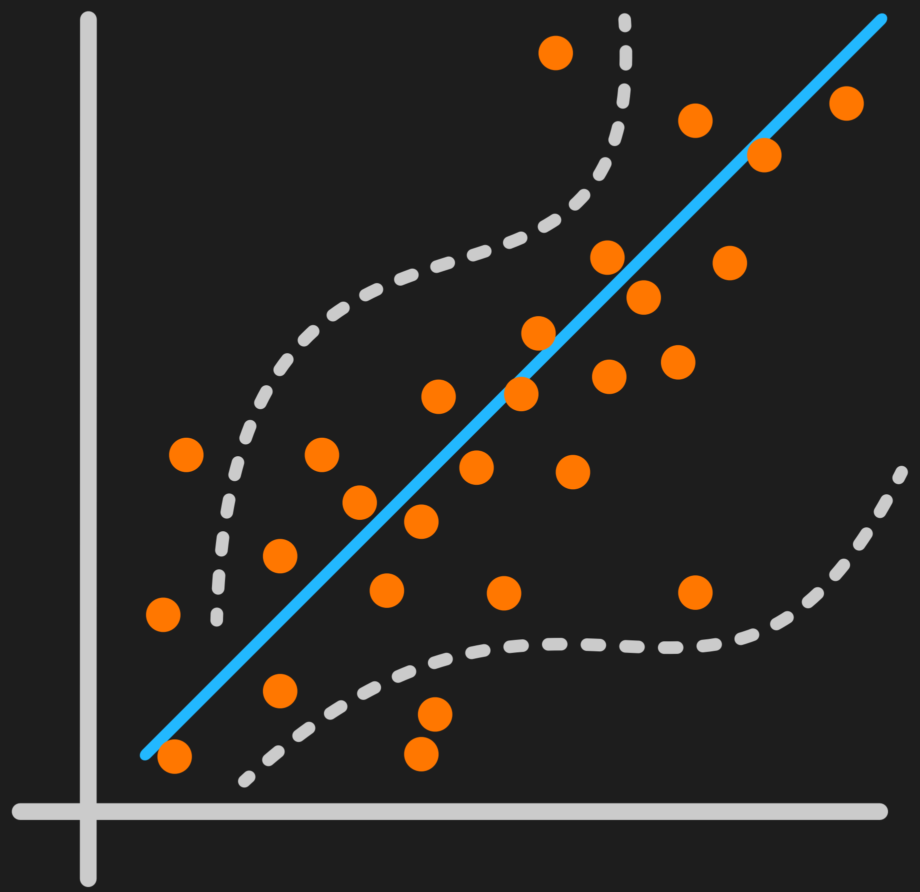
Stakeholder

Real Estate Agent

Prospective Buyer

Property Owner and Developer

Analytics Approach



1. Data Understanding
2. Exploratory Data Analysis
3. Data Preprocessing
4. Modelling
5. Interpretation

Regression Model

Dataset Information

- Each row represents information related to the **one-unit apartment**
- This dataset contains **4123 rows** and **11 columns**
- The oldest apartment was built in **1978** and the newest built in **2015**.

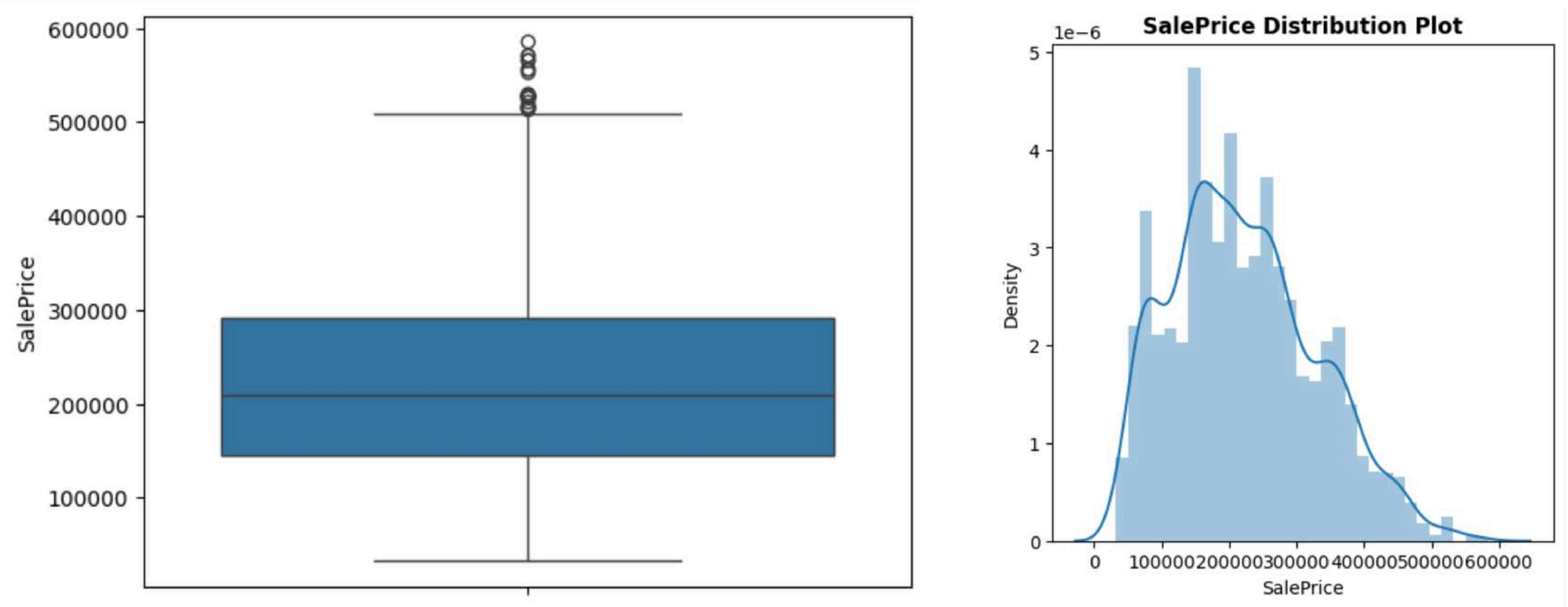
Attribute	Data Type	Description
HallwayType	Object	Types of apartment hallways
TimeToSubway	Object	Measure time takes from apartment to subway station
Subway Station	Object	Name of subway station nearby apartment
N_FacilitiesNearBy(ETC)	Float	number of other facilities such as hotels and special schools
N_FacilitiesNearBy(PublicOffice)	Float	Number of public offices nearby apartment
N_SchoolNearBy(University)	Float	Number of universities nearby apartment
N_Parkinglot(Basement)	Float	Count number of parking spaces on basement
YearBuilt	Integer	The year when the apartment was created
N_FacilitiesInApt	Integer	Number of facilities for residents like swimming pool, gym, play ground
Size(sqf)	Integer	Size of apartment in square feet
SalePrice	Integer	Apartment price in Korean Won (KRW)

Data Checking

	feature	data_type	null	negative	n_nunique	sample_unique
0	HallwayType	object	0.0	False	3	[terraced, mixed, corridor]
1	TimeToSubway	object	0.0	False	5	[0-5min, 10min~15min, 15min~20min, 5min~10min, no_bus_stop_nearby]
2	SubwayStation	object	0.0	False	8	[Kyungbuk_uni_hospital, Chil-sung-market, Bangoge, Sin-nam, Banwoldang, no_subway_nearby, Myung-duk, Daegu]
3	N_FacilitiesNearBy(ETC)	float64	0.0	False	4	[0.0, 1.0, 5.0, 2.0]
4	N_FacilitiesNearBy(PublicOffice)	float64	0.0	False	8	[3.0, 5.0, 7.0, 1.0, 4.0, 2.0, 6.0, 0.0]
5	N_SchoolNearBy(University)	float64	0.0	False	6	[2.0, 1.0, 3.0, 4.0, 5.0, 0.0]
6	N_Parkinglot(Basement)	float64	0.0	False	20	[1270.0, 0.0, 56.0, 798.0, 536.0, 605.0, 203.0, 108.0, 1174.0, 930.0, 475.0, 184.0, 400.0, 218.0, 1321.0, 524.0, 76.0, 79.0, 181.0, 18.0]
7	YearBuilt	int64	0.0	False	16	[2007, 1986, 1997, 2005, 2006, 2009, 2014, 1993, 2013, 2008, 2015, 1978, 1985, 1992, 2003, 1980]
8	N_FacilitiesInApt	int64	0.0	False	9	[10, 4, 5, 7, 2, 9, 8, 1, 3]
9	Size(sqf)	int64	0.0	False	89	[1387, 914, 558, 1743, 1334, 572, 910, 288, 1131, 843, 1160, 644, 829, 743, 868, 1629, 1690, 1273, 1483, 156, 1412, 1394, 903, 676, 355, 1419, 640, 1184, 1167, 135, 818, 206, 1643, 907, 1377, 2337, 1252, 451, 587, 811, 2056, 508, 576, 1366, 1103, 426, 281, 1327, 1092, 857, 1928, 1149, 1088, 1288, 1761, 1437, 1291, 2092, 636, 814, 871, 1519, 1444, 1451, 1448, 1313, 1256, 1796, 1192, 1035, 846, 273, 277, 779, 498, 736, 138, 430, 213, 163, 1369, 192, 547, 839, 160, 793, 1085, 1060, 832]
10	SalePrice	int64	0.0	False	838	[346017, 150442, 61946, 165486, 311504, 118584, 326548, 143362, 172566, 99823, 211504, 305309, 145132, 209734, 168141, 144752, 389380, 347787, 263345, 207079, 149274, 200000, 85132, 245132, 256637, 207964, 371681, 442477, 435398, 75920, 280530, 163716, 263716, 286725, 138938, 57522, 302654, 391150, 215176, 75221, 476106, 241592, 411504, 123008, 115929, 269026, 348672, 295575, 309292, 77876, 345132, 323893, 198230, 372566, 164601, 109734, 247787, 158407, 126548, 146017, 203539, 161946, 183628, 195575, 331858, 138053, 218584, 380530, 277876, 63274, 258079, 231415, 141150, 250176, 56637, 242035, 432743, 274336, 74256, 84955, 147761, 143389, 130973, 79646, 151327, 295460, 72920, 495575, 89380, 353982, 285840, 228318, 469026, 324778, 243362, 343362, 159292, 265486, 318584, 460176, ...]

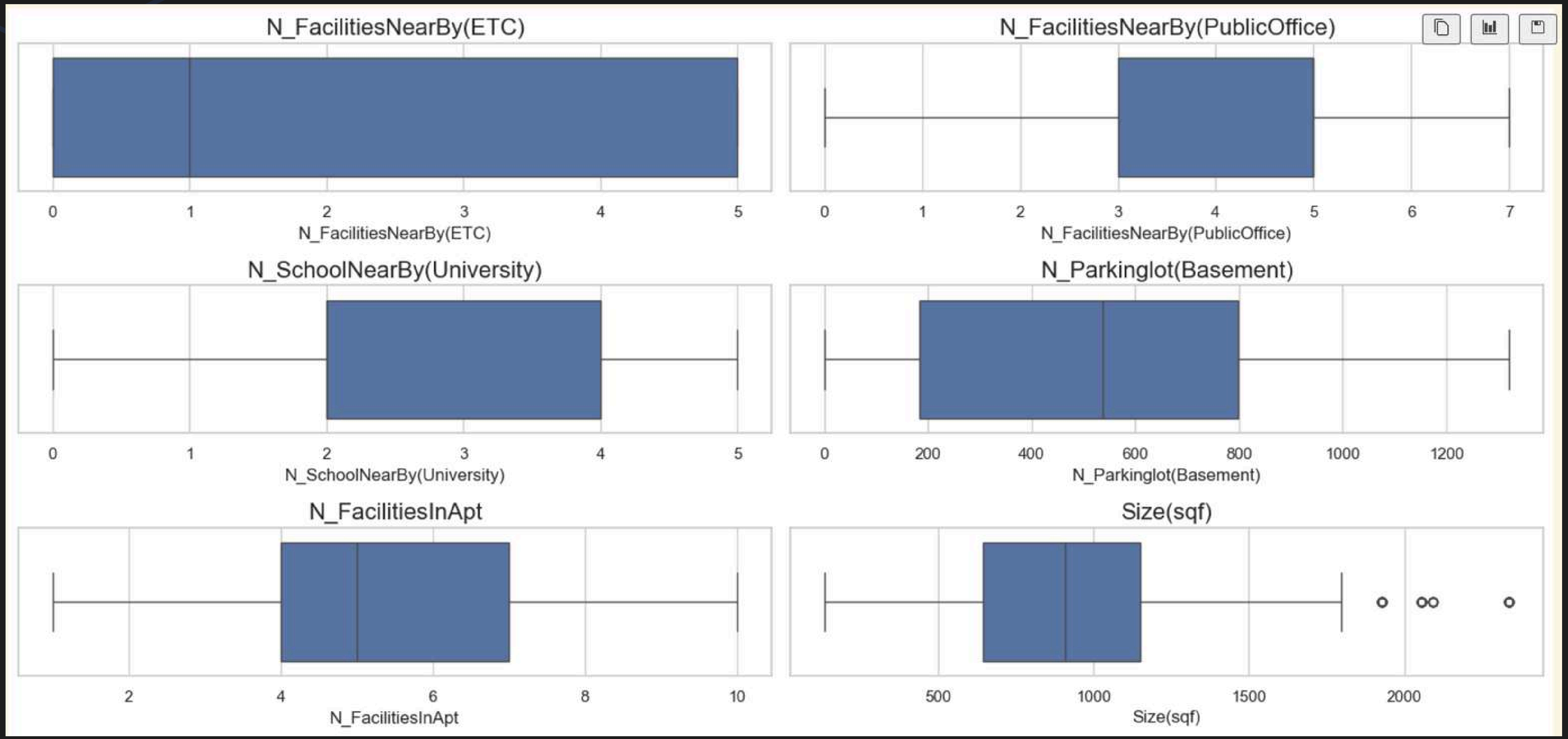
- No null values
- No negatives values

Distribution of Target



- Not normal distribution
- There are outliers

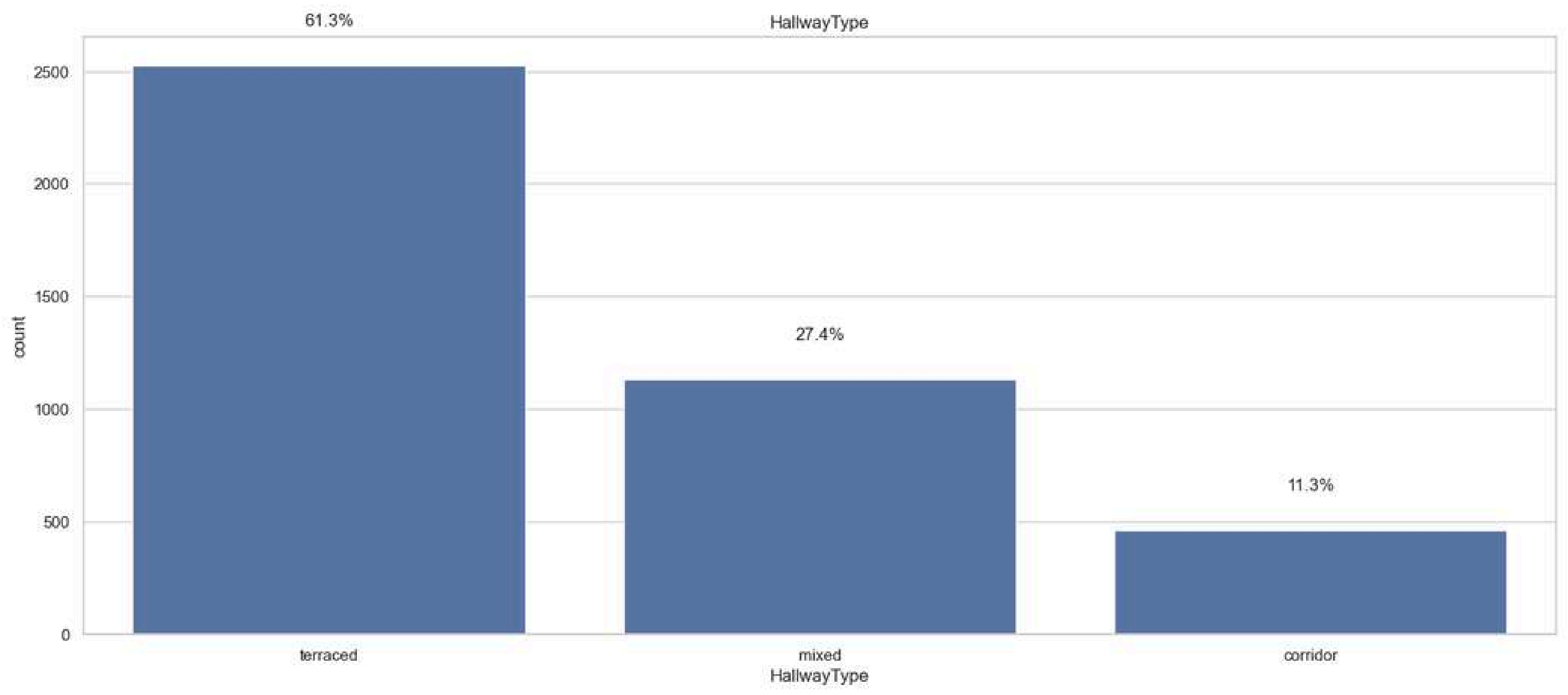
Distribution of Numerical Feature

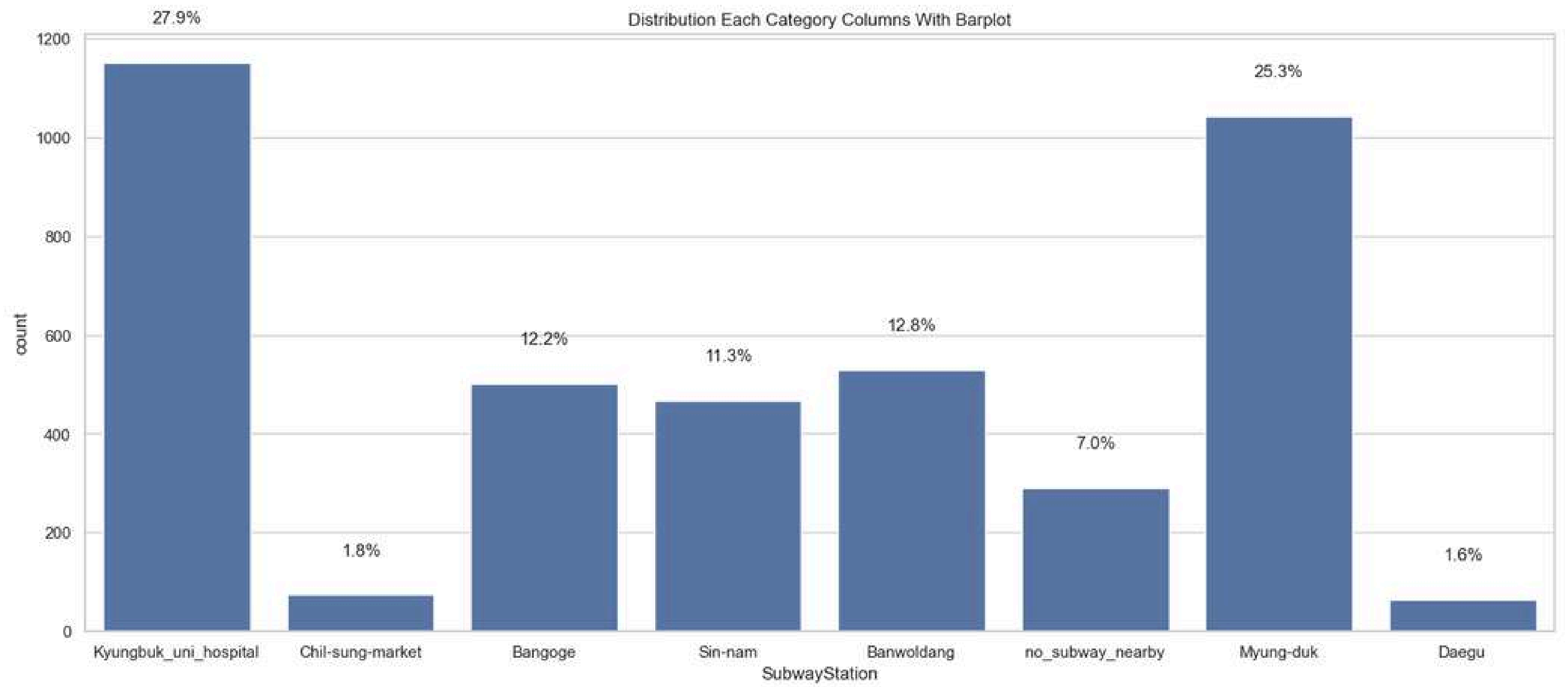


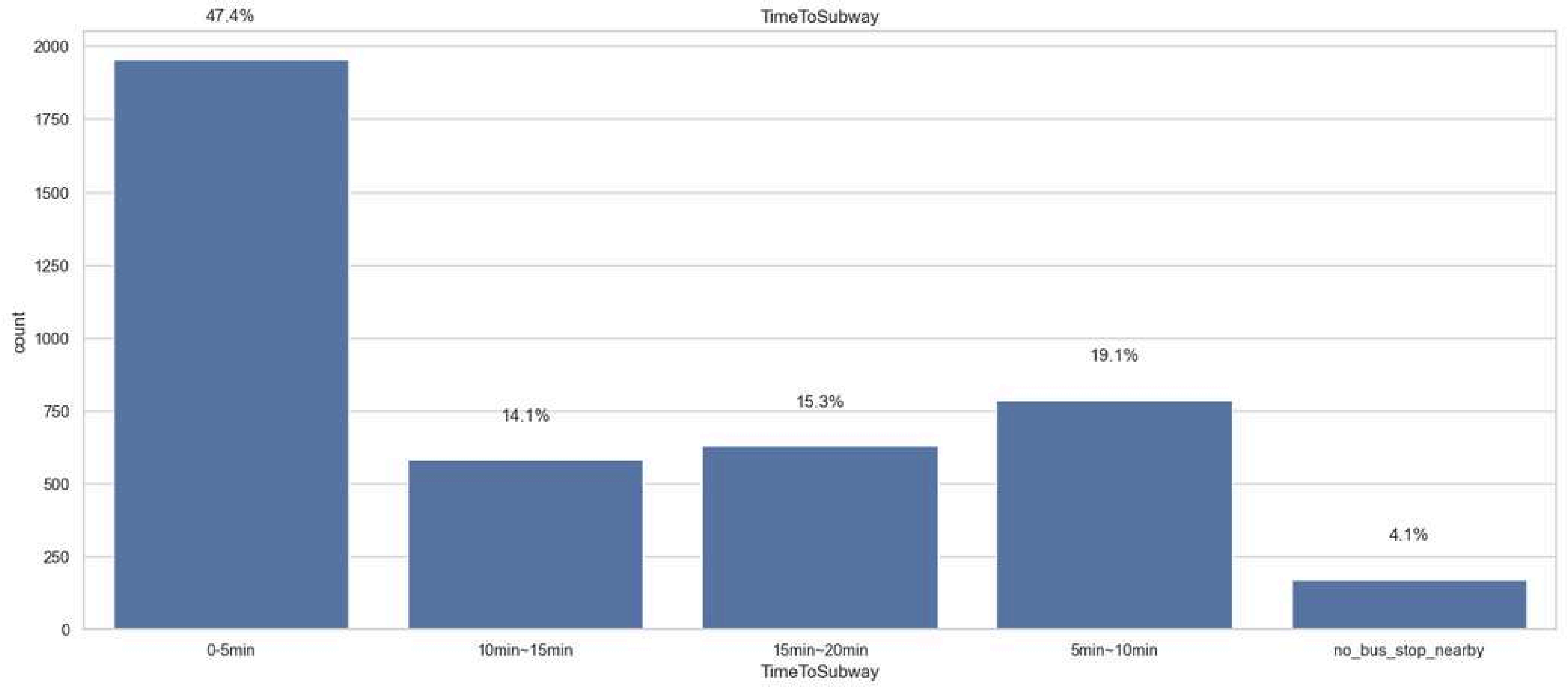
- There are outliers in Size(sqf)

Distribution of Categorical Feature

- Most apartment in Daegu has terraced hallway type
- Most apartment in Daegu is built near to subway
- Most apartment in Daegu close to Kyungbuk Uni Hospital Subway Station



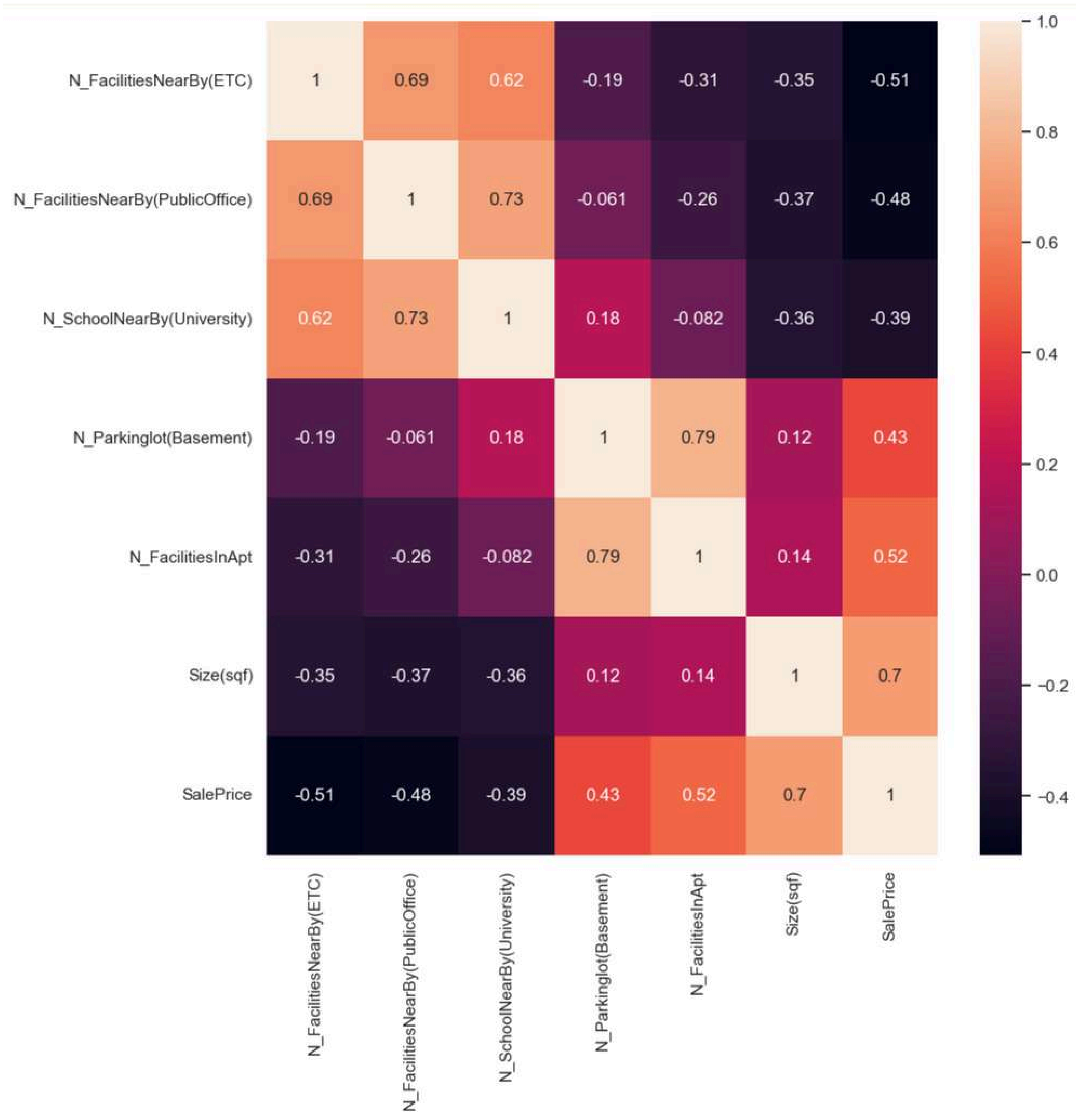




Correlation between feature

The heatmap indicates a significant positive correlation between the following variables:

- **'University'** with **'PublicOffice'**
- **'Facilites Apart'** with **'Basement'**
- **'SalePrice'** with **'Size'**



Detect Anomaly

	HallwayType	TimeToSubway	SubwayStation	N_FacilitiesNearBy(ETC)	N_FacilitiesNearBy(PublicOffice)	N_SchoolNearBy(University)	N_Parkinglot(Basement)	YearBuilt	N_FacilitiesInApt	Size(sqf)	SalePrice
37	corridor	5min~10min	no_subway_nearby	1.0	4.0	1.0	218.0	2014	1	156	57522
39	terraced	5min~10min	no_subway_nearby	0.0	1.0	1.0	1321.0	2015	10	910	391150
44	terraced	5min~10min	no_subway_nearby	0.0	1.0	1.0	1321.0	2015	10	914	411504
83	corridor	5min~10min	no_subway_nearby	1.0	4.0	1.0	218.0	2014	1	135	56637
165	terraced	5min~10min	no_subway_nearby	0.0	1.0	1.0	1321.0	2015	10	644	256637
...	Execution Order	...	...	...	...	...	...	...	...	...	...
3818	terraced	5min~10min	no_subway_nearby	0.0	1.0	1.0	1321.0	2015	10	644	256637
3836	terraced	5min~10min	no_subway_nearby	0.0	1.0	1.0	1321.0	2015	10	644	252212
3841	terraced	5min~10min	no_subway_nearby	0.0	1.0	1.0	1321.0	2015	10	910	394690
3886	terraced	5min~10min	no_subway_nearby	0.0	1.0	1.0	1321.0	2015	10	644	269911
3997	terraced	5min~10min	no_subway_nearby	0.0	1.0	1.0	1321.0	2015	10	910	317699

119 rows × 11 columns

During the analysis, I found there're an anomaly for about **119** in `**TimeToSubway**` equal to **5min~10min** eventhough value in column `**SubwayStation**` state **no_subway_nearby.**

Data Cleaning

- Change Data Type
- Handling Inconsistency
- Handling Anomaly
- Handling Duplicated
- Handling Outliers

Data Transformation

Tree-based Model

- **One-hot encoder** for categorical features

Linear Model

- **One-hot encoder** for categorical features
- **Standard Scaling** for Numerical features
- **Feature Engineering** using Polynomial Feature

Feature Selection

- Doing F-test to assess the **statistical significance** of each feature

Model Benchmarking

Model	Description
Lasso Regression	Lasso (Least Absolute Shrinkage and Selection Operator) Regression performs L1 regularization, which can shrink some coefficients to zero, thus performing variable selection and regularization simultaneously. This helps in handling multicollinearity and reducing the complexity of the model.
Ridge Regression	Ridge Regression applies L2 regularization, which penalizes the size of the coefficients. This model helps to prevent overfitting by shrinking the coefficients, but unlike Lasso, it does not set any coefficients to zero. It's useful when dealing with multicollinearity.
Random Forest Regression	An ensemble learning method that constructs multiple decision trees during training and outputs the average of the predictions of the individual trees. It reduces overfitting and improves accuracy by combining the predictions of several trees.
XGBoost Regression	Extreme Gradient Boosting (XGBoost) is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost uses a more regularized model formalization to control overfitting, leading to better performance.
Decision Tree Regression	A non-linear regression model that splits the data into subsets based on feature values. It constructs a tree where each node represents a feature, each branch represents a decision rule, and each leaf represents an outcome. It can capture non-linear relationships but is prone to overfitting.
Extra Trees Regression	Extra Trees (Extremely Randomized Trees) Regression is similar to Random Forest but differs in how the trees are constructed. Extra Trees use the whole dataset and randomly select the cut points for each feature. This results in more variability and can reduce overfitting.
Stacking Regressor	Stacking Regressor is an ensemble learning technique that combines multiple regression models (base models) to improve predictive performance. A meta-model is trained on the predictions of the base models to provide a final prediction. It leverages the strengths of multiple algorithms. This is combination between Linear Regression, Ridge, Lasso, and Random Forest
Ordinary Least Squares (OLS) Regression	OLS Regression is a method for estimating the unknown parameters in a linear regression model. It minimizes the sum of the squared differences between observed

Adjusted R2

RMSE

Splitting with 80% and 20%

Model Benchmarking

	Name	RMSE Train	RMSE Test	Difference RMSE	Adjusted R² Train	Adjusted R² Test
6	Stacking Regressor	40881.613734	41205.447535	-323.833801	0.841688	0.843055
2	Random Forest	41507.715511	41693.168685	185.453174	0.836496	0.838100
3	Extreme Gradient Boosting	41397.744067	41795.898826	398.154758	0.837361	0.837301
5	Extra Tree	41397.742935	41823.283383	425.540448	0.837361	0.837088
4	Decision Tree	41397.742935	41827.258233	429.515298	0.837361	0.837057
7	OLS	49791.832159	50581.716567	789.884408	0.764498	0.760805
0	Lasso	53296.783053	54062.456505	765.673452	0.730429	0.727787
1	Ridge	53296.913601	54069.607304	772.693703	0.730427	0.727715

Stacking Regressor

Random Forest

Hyperparameter Tuning

Random Forest		
Parameter	**Description**	**Value**
n_estimator	controls the number of decision trees in the forest.	200
max_depth	determines the maximum depth of each individual tree.	10
min_sample_split	defines the minimum number of samples required to split a node into daughter nodes.	10
min_sample_leaf	sets the minimum number of samples required to be at a leaf node.	1
bootstrap	determines whether to use bootstrapping with replacement during tree building	False

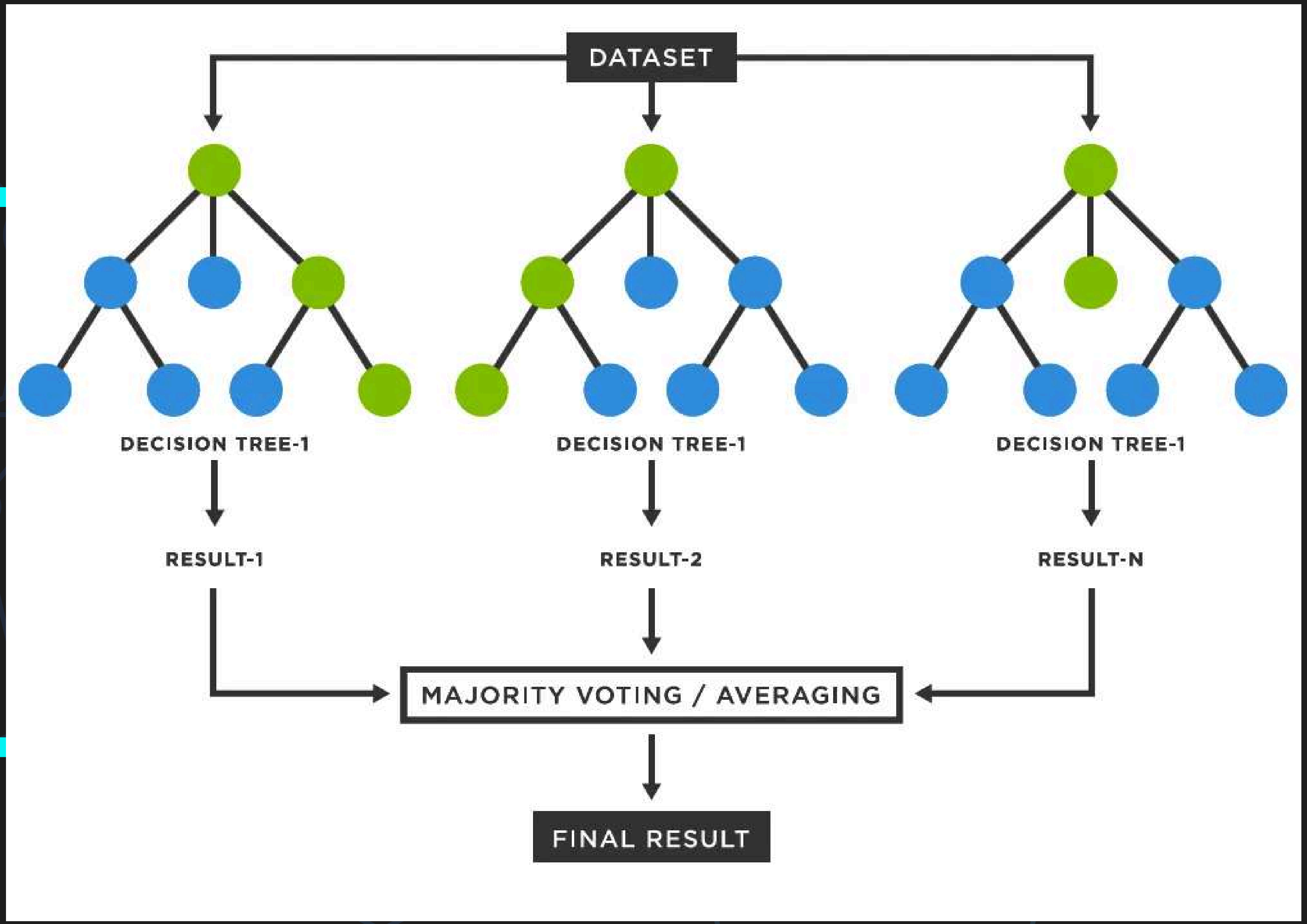
Stacking Regressor			
Model	**Parameter**	**Description**	**Value**
Ridge	alpha	controls the strength of the regularization penalty in the ridge regression component of the Stacking ensemble.	10
Random Forest	max_depth	determines the maximum depth of each individual tree.	10
Random Forest	n_estimator	determines the number of trees to grow in the random forest, potentially impacting model complexity and accuracy.	300
Linear Regression	fit_intercept	controls whether the linear regression component in the Stacking ensemble should fit an intercept term.	False
Lasso	alpha	controls the strength of the L1 regularization penalty in the Lasso regression component of the Stacking ensemble	0.1
Final Estimator	fit_intercept	controls whether the final regressor in the Stacking ensemble (Linear Regression) should fit an intercept term.	False

Model Evaluation

	Model	RMSE Train	RMSE Test	RMSPE Train	RMSPE Test	Adj R2 Test	Difference RMSE
2	Stacking Regressor (Initial)	40881.613734	41205.447535	22.684923	22.800072	0.843055	323.833801
0	Random Forest (Initial)	41504.711004	41726.174294	25.038891	23.407702	0.839063	221.463290
1	Random Forest (Tuned)	41459.080814	41765.879023	24.898011	23.469854	0.837535	306.798209
3	Stacking Regressor (Tuned)	59754.995147	51196.257316	54.298570	28.455374	0.757721	-8558.737831

After evaluating the models' performance, **Random Forest Initial** was chosen over Stacking Regressor due to:

- *RMSE Test 41726.1743*
- *Adjusted R2 0.839*
- *Interpretability*
- *RMSPE Test 23.4%*
- *Difference RMSE 221.4633*
- *Model Stability*

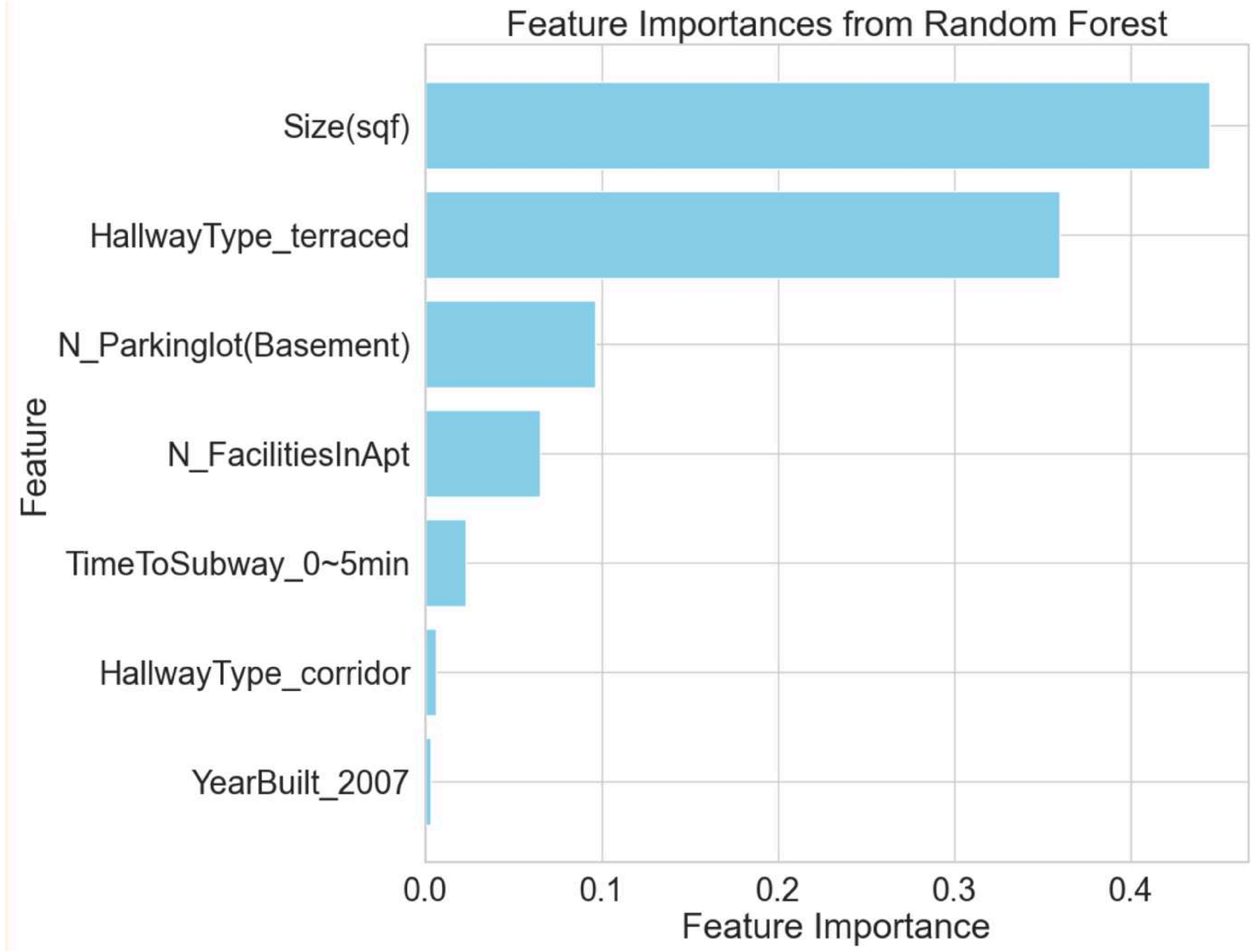


How Random Forest Work

1. Sample with Replacement (Bootstrapping)
2. Building Decision Trees
3. Forest of Diverse Trees

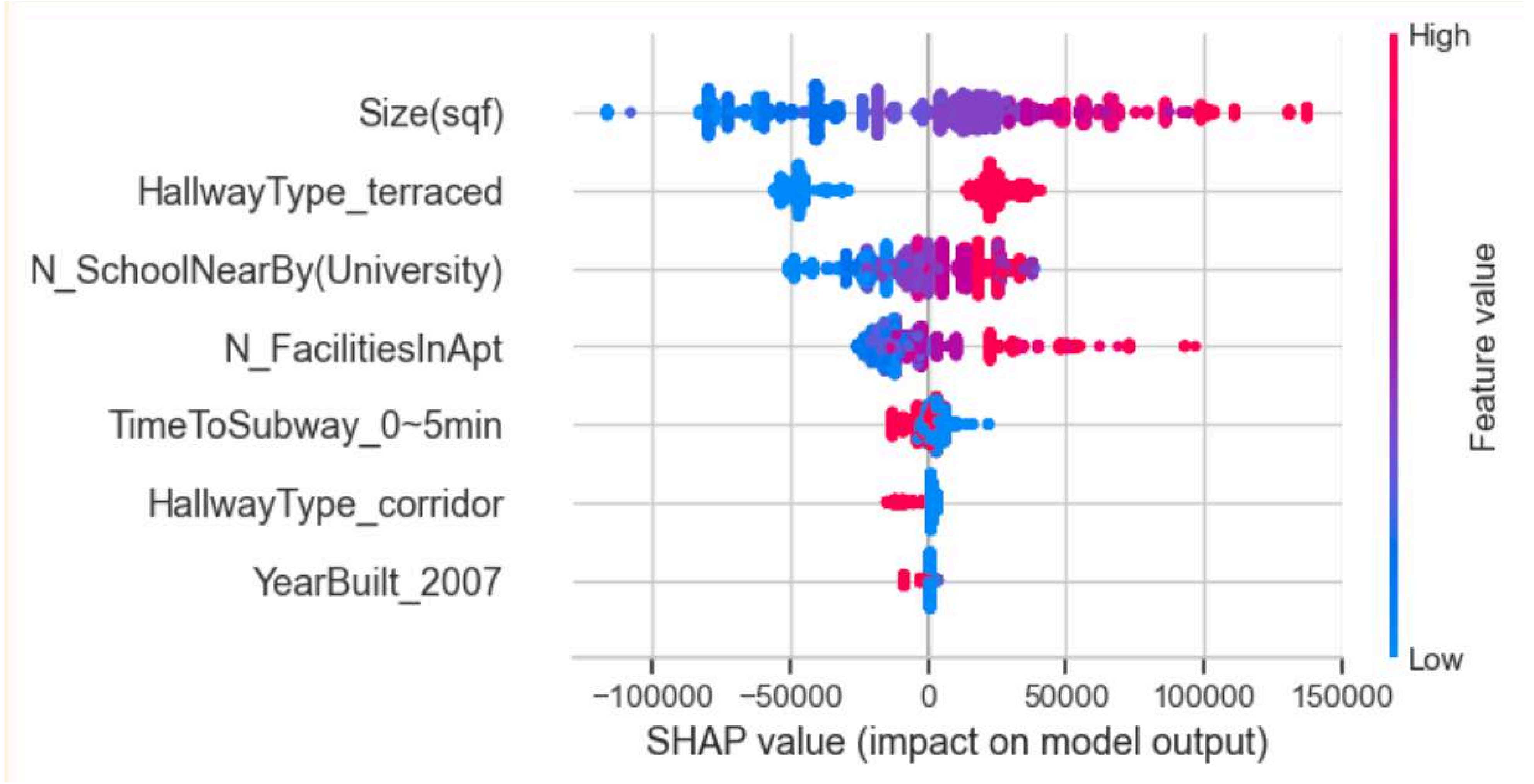
Gini Importance

Size(sql), hallway type terraced, and **basement parking** are the top three features influencing apartment prices.



SHAP
(Shapey Value of Explanations)

- **Size(sqf), HallwayType_terraced and University** are the most influential features in predicting apartment prices in Daegu.



Addressing Business Problem

This model can have several positive impacts if it deployed to address business problem:

Market Stabilization

Informed Decision Making

Affordability for Buyer

SHAP

Conclusion

Best Model

Random Forest Initial with **RMSE(41,726), RMSPE(23,41%)** and **Adjusted R^2 0.839**

RMSE

The model's predictions deviate from the actual apartment prices by approximately **₩41,726.17**

Gini Importance

- Size (sqf)
- Hallway Type Terraced
- Basement parking

SHAP

- Size (sqf)
- Hallway Type Terraced
- University

Recommendation

Additional Feature

Incorporate additional features such as **economic indicators, interest rates,** and **government policies** that might affect housing prices.

Comprehensive Dataset

Integrate more comprehensive datasets, including **demographic information**, **proximity to amenities,** and **historical price trends,** to improve the model's predictive power.

Explore Other Model

Try other regression algorithms such as Neural Network to get any other comparison within the model

Limitation

Price Range

Range of Sale Prices in the training data, which is between **₩32,743 and ₩508,849**

Economic

The model does not account for macroeconomic factors such as **interest rates, economic growth, or government policies**

Feature Focus

The model primarily focuses on features such as

- **Size(sqf)**
- **HallwayType**
- **N_Parkinglot(Basement)**
- **N_FacilitiesInApt**
- **TimeToSubway**
- **YearBuilt**

Market Change

The model assumes that the relationships between features and Sale Price remain **constant over time**

Continue to Cloud

ndaffaabdurahman@gmail.com

Portfolio

LInkedin