House Price Analysis with Machine Learning Approaches

Zishan Shao

2023-03-11

Abstract

This report presents a study on the prediction of house prices using the Boston house saleprice dataset. The aim of this study is to investigate the relationship between various factors that affect house prices and to develop models that can accurately predict the sale prices of houses in the Boston area.

The dataset was cleaned, missing values were imputed, and one-hot encoding was performed on the categorical features. Three different regression models were then trained on the preprocessed dataset: Linear Regression, Lasso Regression, and Random Forest. The performance of these models was evaluated using mean squared error (MSE) and R-squared (R2).

Our results indicate that Random Forest outperforms the other two models, achieving an MSE of 9.38 and an R2 score of 0.88 on the test dataset. We also found that the most important features in predicting house prices were the number of rooms, the percentage of lower status population, and the weighted distance to five Boston employment centers.

This study has important implications for both buyers and sellers in the Boston real estate market, as it provides insights into the factors that drive house prices and the accuracy of different prediction models. The methodology used in this study can also be applied to other real estate datasets to develop accurate prediction models.

Overall, this report contributes to the field of real estate by providing a comprehensive analysis of the Boston house saleprice dataset and demonstrating the effectiveness of various regression models in predicting house prices.

Contents

Introduction

The real estate market is a complex and dynamic system, influenced by a multitude of factors such as location, size, age, and condition of the property. Predicting the price of a house accurately is a crucial task for buyers, sellers, and real estate agents alike. The emergence of big data and machine learning algorithms has brought significant advances in the field of real estate valuation, enabling more precise and data-driven price predictions. In this report, we present our analysis of the housing market in Boston, using various machine learning techniques to predict house prices. We discuss the data collection and preprocessing steps, feature engineering techniques, and model selection process, and evaluate the performance of different models using appropriate metrics. Our results demonstrate the effectiveness of machine learning algorithms in predicting house prices and provide insights for real estate professionals and homeowners.

Data Preprocessing

To ensure that the model achieves expected performance, the data pre-processing is necessary to ensure the assumptions of the data was solid, for instance, we assume that the data has no missing values. However, the training data includes missing values, weird types, and incorrect types of data, so we need to correct those by analyzing the dataset, droping or imputing the missing values, and adjusting the data to a correct format for processing. Thus, we will have data set that fulfills the assumptions of the models.

The training data consists of 1460 observations and 80 features, and the response variable is the sale price of the house. We assume that there is some features of the house that provide us insight of a high house price. The goal of this report is to explore which feature provides the most evidences to the sale price of the house in Boston, and uses the model to relative accurately predict the house price.

Label Processing

Label or target variable processing is an essential step in the data preprocessing pipeline to ensure the accuracy and precision of machine learning models. Skewed or weirdly distributed data can have significant impacts on the quality of the predictions. Skewed data refers to data that has a disproportionate number of values on one side of the distribution compared to the other. For instance, house price data can be skewed if there is a large number of houses with low prices and only a few houses with high prices or vice versa.

The skewness of data could have a significant influence on the accuracy of prediction. For instance, the linear regression models assume that the target variable is normally distributed, and if the assumption was not met, it can lead to biased estimates, incorrect standard errors, and inaccurate predictions. To address this issue, non-parametric methods such as decision trees, random forests, and support vector machines can be used. These methods do not rely on the assumption of normality and can be more effective in handling skewed data. Therefore, it is crucial to check the distribution of the target variable and use appropriate methods to handle it in order to achieve accurate and precise machine learning models.

The following histogram provides the distribution of the house price, indicating there is skewness existed in the target variable, we fix the problem by log transform the response variable.

Figure 1.1 (1):

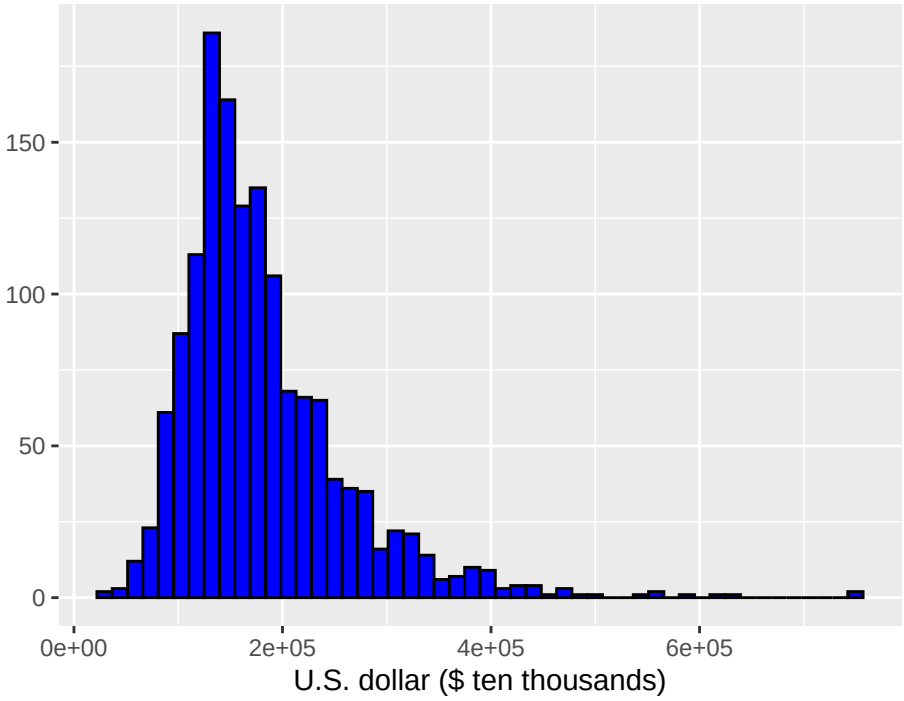
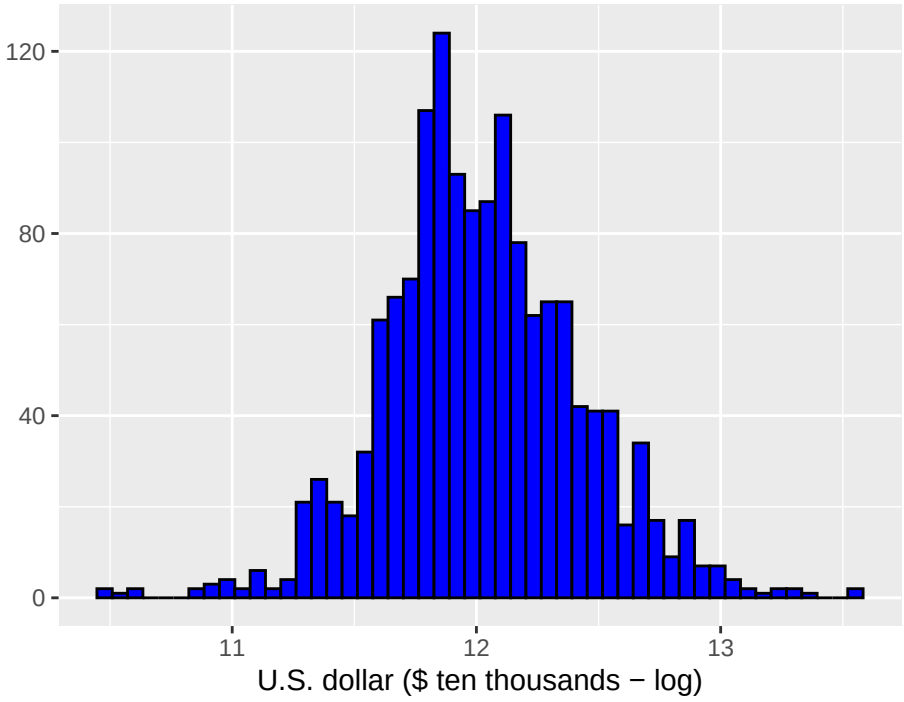


Figure 1.1 (2):



categorical and numerical variables

In this section, we have dealt with the issue of mixed data types in the dataset. We found that there were categorical variables, character variables, and numerical variables in the dataset. We identified the categorical variables by selecting variables that were factors, character variables by selecting variables that were not factors but had a character data type, and numerical variables by selecting variables that had a numeric data type.

We noticed that the categorical variables in the dataset were mostly represented by character variables rather than factors. Therefore, we converted all the character variables to factors using the built-in R function. This allowed us to transform the character variables into categorical variables and make sure that they were represented as factors, which is important for many statistical analyses and machine learning models.

After the conversion, we checked the data type of the variables, confirming that all character variables had been converted to factors. We found that the dataset now only contained categorical variables and numerical variables, with all the categorical variables represented as factors. This ensures that the dataset is now ready for further analysis, as we have transformed all the variables into a consistent and usable format.

It is worth mentioning that, although data can be broadly classified into categorical and numerical, there are more than these two broad categories. Categorical data can be further divided into two subtypes: nominal and ordinal. Nominal data are those where data points are named or labeled and are similar to a noun, for example, a person's name or gender. Ordinal data is ranked, ordered, or used on a rating scale, such as a condition score or quality ranking.

Numerical data can be divided into three subtypes: continuous, discrete, and interval data. Continuous numerical data represent measurements and their intervals fall on a number line, so it doesn't involve taking counts of the items. Discrete data is used to represent countable items, and can take both numerical and categorical forms and group them into a list. This list can be finite or infinite. Interval data is a type of numerical data that can be measured only along a scale at equal distances from each other. The numerical values in this data type can only undergo add and subtract operations.

Understanding the data type of each variable in a dataset is crucial because different statistical methods and models require different types of data. It is also important to properly transform and preprocess the data before modeling to ensure accurate and reliable results.

	data_type
Id	integer
MSSubClass	integer
MSZoning	factor
LotFrontage	integer
LotArea	integer
Street	factor
Alley	factor
LotShape	factor
LandContour	factor
Utilities	factor
LotConfig	factor
LandSlope	factor
Neighborhood	factor
Condition1	factor
Condition2	factor
BldgType	factor
HouseStyle	factor
OverallQual	integer
OverallCond	integer
YearBuilt	integer

	data_type
YearRemodAdd	integer
RoofStyle	factor
RoofMatl	factor
Exterior1st	factor
Exterior2nd	factor
MasVnrType	factor
MasVnrArea	integer
ExterQual	factor
ExterCond	factor
Foundation	factor
BsmtQual	factor
BsmtCond	factor
BsmtExposure	factor
BsmtFinType1	factor
BsmtFinSF1	integer
BsmtFinType2	factor
BsmtFinSF2	integer
BsmtUnfSF	integer
TotalBsmtSF	integer
Heating	factor
HeatingQC	factor
CentralAir	factor
Electrical	factor
X1stFlrSF	integer
X2ndFlrSF	integer
LowQualFinSF	integer
GrLivArea	integer
BsmtFullBath	integer
BsmtHalfBath	integer
FullBath	integer
HalfBath	integer
BedroomAbvGr	integer
KitchenAbvGr	integer
KitchenQual	factor
TotRmsAbvGrd	integer
Functional	factor
Fireplaces	integer
FireplaceQu	factor
GarageType	factor
GarageYrBlt	integer
GarageFinish	factor
GarageCars	integer
GarageArea	integer
GarageQual	factor
GarageCond	factor
PavedDrive	factor
WoodDeckSF	integer
OpenPorchSF	integer
EnclosedPorch	integer
X3SsnPorch	integer
ScreenPorch	integer
PoolArea	integer

	data_type
PoolQC	factor
Fence	factor
MiscFeature	factor
MiscVal	integer
MoSold	integer
YrSold	integer
SaleType	factor
SaleCondition	factor
SalePrice	numeric

Missing data handling

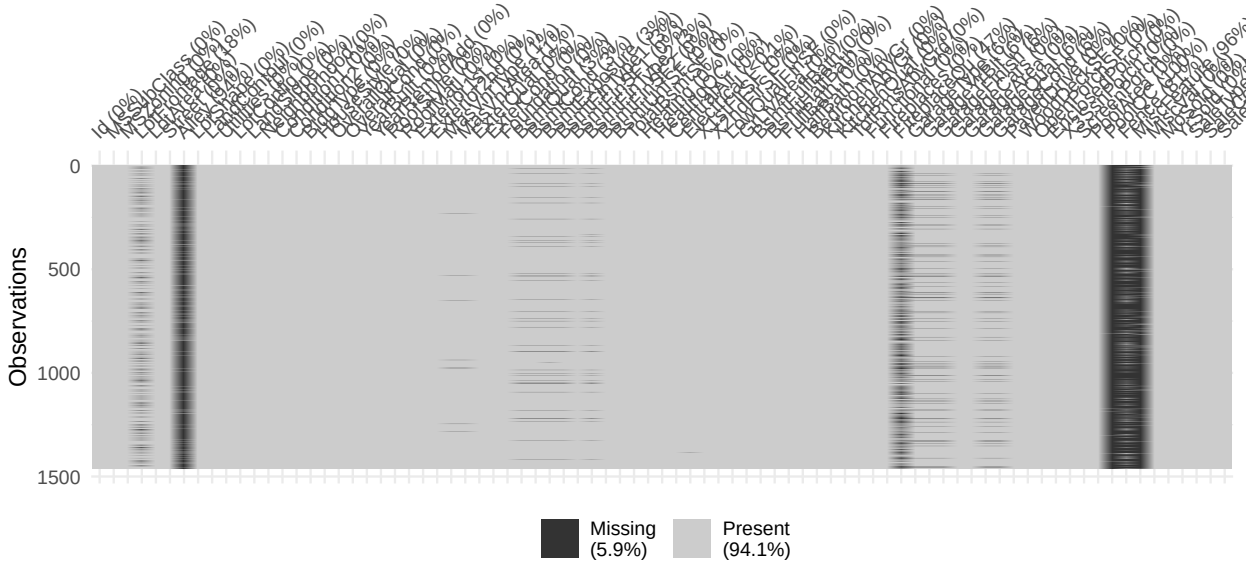
Handling missing values is a crucial step in any data analysis tasks. Because missing values can arise due to a variety of reasons, such as human error, incomplete data entry, or data corruption, so it can have a significant impact on the accuracy and validity of statistical analyses and predictive models.

Ignoring or mishandling missing values can lead to biased or incorrect conclusions, as well as reduced power and precision in statistical tests. Missing values can also introduce noise and variability into models, which can negatively affect their performance and generalizability.

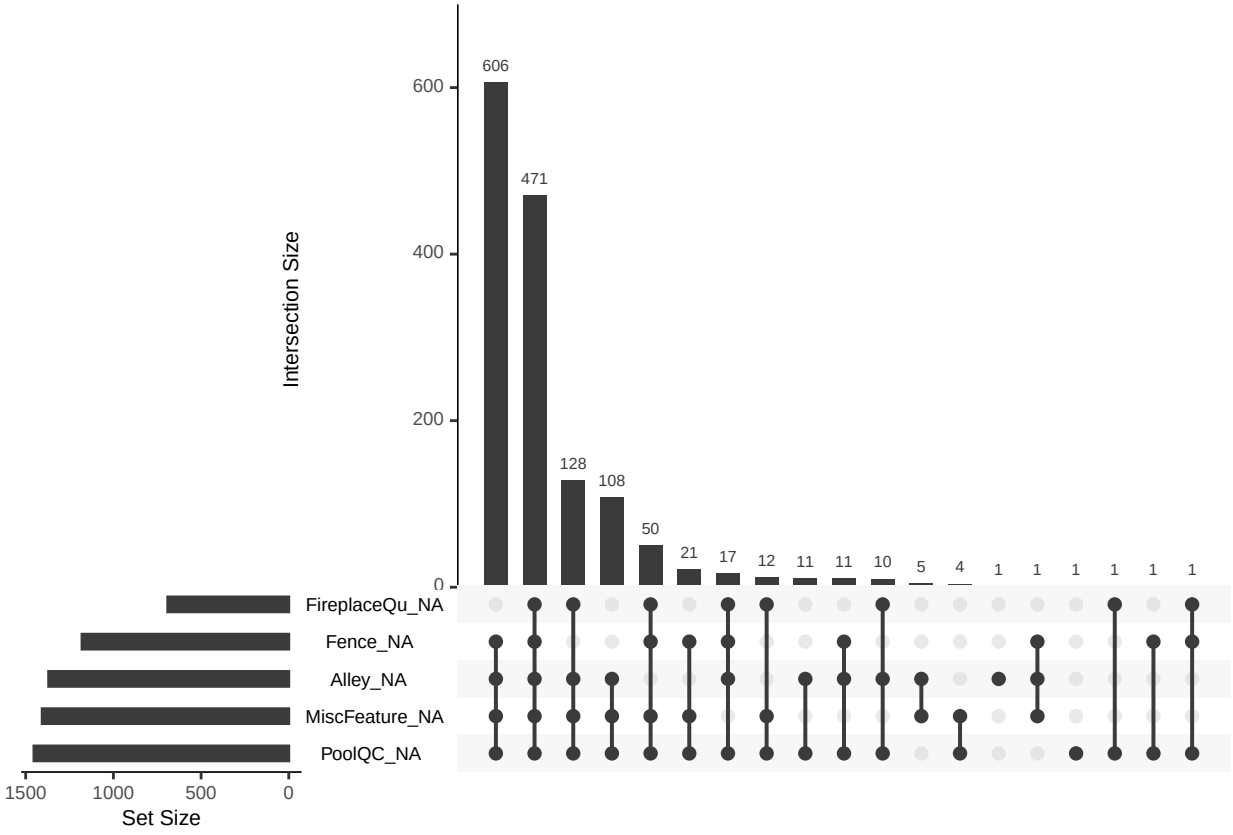
Thus, it is essential to have a clear strategy for handling missing values in any data analysis project. This strategy should involve identifying missing values, understanding their distribution and potential causes, and selecting appropriate methods for imputing or handling missing values. By properly handling missing values, we can ensure the accuracy and validity of our analyses and models, as well as maximize the value of our data.

In this section, we will firstly identify where the outliers lay by visualizing the missing values in each feature, and then identify the proper method to handling the missing values. Finally, we will summarize and apply the cleaned dataset.

Part 1: Visualization



we found that there are a number of variables with a large number of missing values, including Fireplace, Fence, Alley, MiscFeature, and Pool. The PoolQC, for instance, has almost 100% missing values.



What's more, there are many intersecting missing values, which indicates that these values may missing for reasons, such as certain types of house does not have certain features, so it is unreasonable to conduct complete case analysis, which is removing all observations with missing values. Therefore, it is important to investigate the reasons for missingness and handle the missing data appropriately with imputations, as it can impact the overall analysis of the data.

Table 2: Missing Values Counting (Categorical Variables)

	Count of Missing (%)
Alley	93.7671233
MasVnrType	0.5479452
BsmtQual	2.5342466
BsmtCond	2.5342466
BsmtExposure	2.6027397
BsmtFinType1	2.5342466
BsmtFinType2	2.6027397
Electrical	0.0684932
FireplaceQu	47.2602740
GarageType	5.5479452
GarageFinish	5.5479452
GarageQual	5.5479452
GarageCond	5.5479452
PoolQC	99.5205479

	Count of Missing (%)
Fence	80.7534247
MiscFeature	96.3013699

Table 3: Missing Values Counting (Numerical Variables)

	Count of Missing (%)
LotFrontage	17.7397260
MasVnrArea	0.5479452
GarageYrBlt	5.5479452

Based on the summary table of missing values, we found that the missing values in the qualitative data share some prefix such as "Garage" or "Bsmt", so these could be the houses that does not have the basement or garage.

Part 2: Evaluation and imputation

Because variables are missing for reason, we decided to impute the value rather than removing all of them. For qualitative variables, we decided to make the missing values as a separate category as they are missing for a reason, which means these houses may share a similar pattern in their house price.

For the numerical variables, the situation was more complicated because the values are continuous and could not be treated as a separate category. We decided to impute the values by situation. For instance, we impute the lot frontage with the mean of the houses in the local neighborhood, as we know that the neighborhood data has no missing values.

To make our model more interpretable, we also need to preprocess the features. For instance, the categories of the Neighborhoods has awkward names, so we replace them with serial numbers. We also noticed that some of the features are, like the target, also skewed, so we correct those with the skewness score greater than 0.8.

Models

In this section, three types of model have been built: least-squares linear regression (LSLR) model, decision tree, and random forest. All three models are applied to explore the factors that post most significant influence on the house sales price, and their prediction accuracy was also measured by the Mean-Squares Error (MSE), one of the most widely used performance measuring metric for regression tasks in Machine Learning.

The mean squares error is a measure of the average squared difference between the predicted values and the actual values. It is given by the formula:

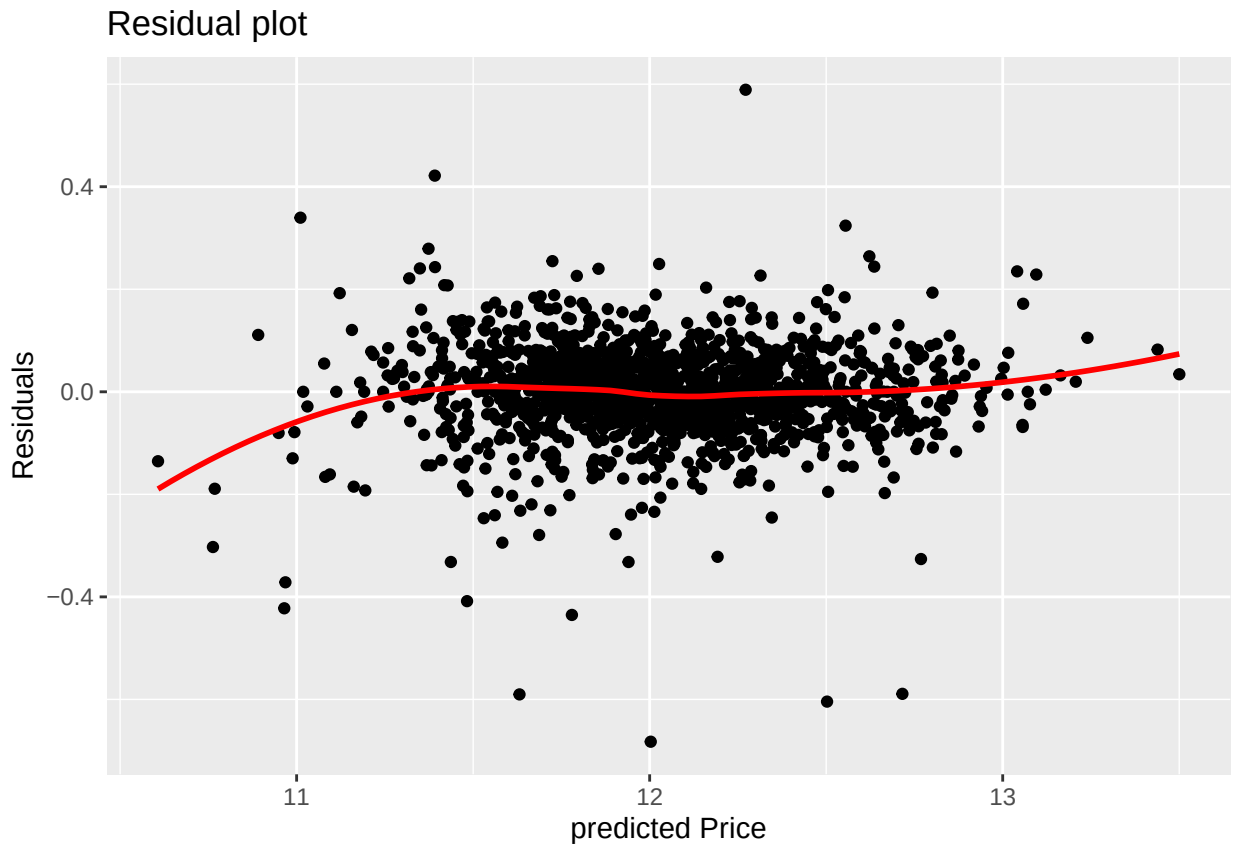
$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

For the association tasks, we uses the association variables such as adjusted R squares, variable importance, to understand the importance of each variable in explaining the sales price of the house.

Least-squares linear regression (LSLR):

For the LSLR, we firstly fit a null model to find the null deviance. The default LSLR model contains only the intercept and the predicted value is the average sales price of all observations. Based on the output of the model, the null deviance is 232.8. The deviance meansures the variance of the model remaining unexplained and was an indicator of the R squared.

The next step is to fit the model with all variables, which provides us with an adjusted R-squared of 0.9366, and deviance of 12.193, indicating that the variables explains total 93.7% of variance of the model, and the deviance has decreased significantly. The residual plot indicates the residuals are mostly zero-meaned, so our model has correctly capture the pattern of the data.



However, we have also noticed that there are some insignificant variables (with p-value > 0.05 for the t-test), which means there are some variables that does not help explaining the sales price. In this case, we choose some of the significant factors from the previous model to see its performance.

In this case, our subset model was as following variables: MSZoning + LotArea + Neighborhood + OverallQual + OverallCond + YearBuilt + GrLivArea + KitchenQual

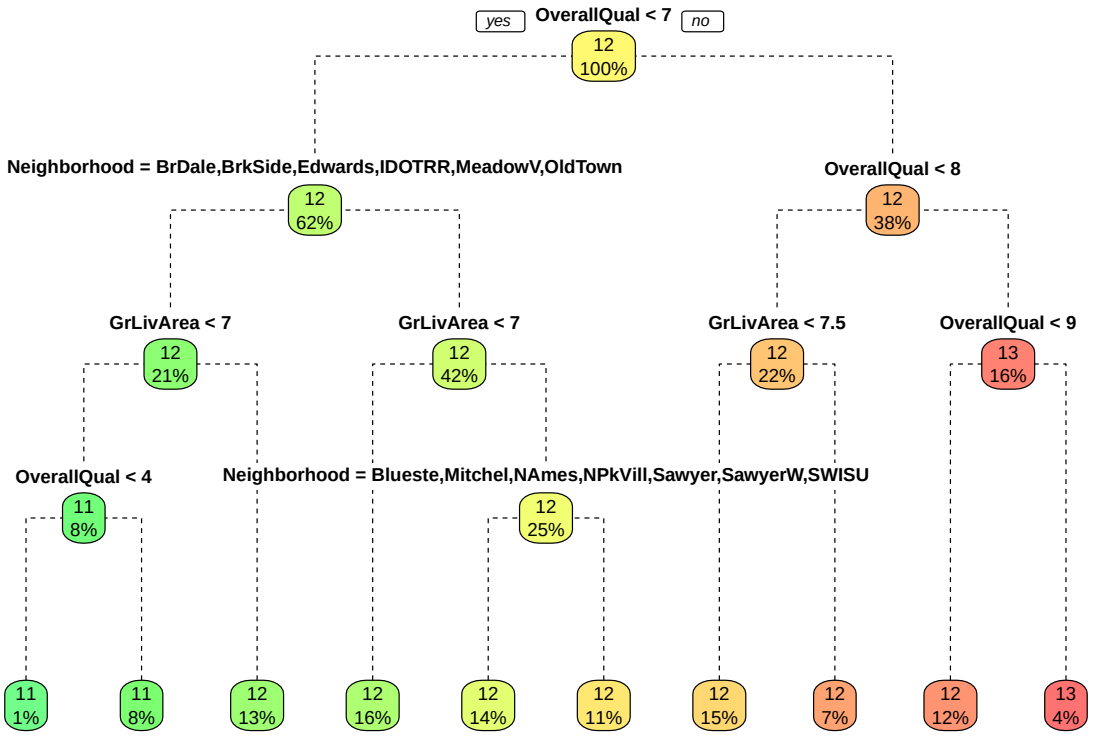
We then fit the subset model and explore its performance. The smaller model has achieved a higher adjusted R-squared (0.8756) and a slightly increased MSE to 28.24. However, the number of predictor used was signficantly less than the full model. Therefore, the variables in subset model provides evidences to the sales price of the house.

It is worth mentioning that the test MSE is unable to be derived because there is correlated features in the dataset. If two variables are closely related with each other, the LSLR could not give a valid prediction. Therefore, we use another powerful method that can effectively handling the complicated relationship in dataset.

Decision Tree

A decision tree is a popular supervised learning algorithm that builds a tree-like model of decisions and outcomes, based on recursively splitting the data into subsets using the most informative features. In this case, we build the decision tree with all features in the training dataset. Becasuse the decision tree split on the variable that provides the highest information gain, so we can explore the variables that is most closely related with the house sales price based on the order of the splitting rule for the nature of the decision tree.

The splitting rule indicates that the overall quality, neighborhood, ground living areas provides strong evidence in explaining the sales price of the house. Noticing that these variables are alos variables that is test significant in the LSLR model (for categorical variable, at least some levels).



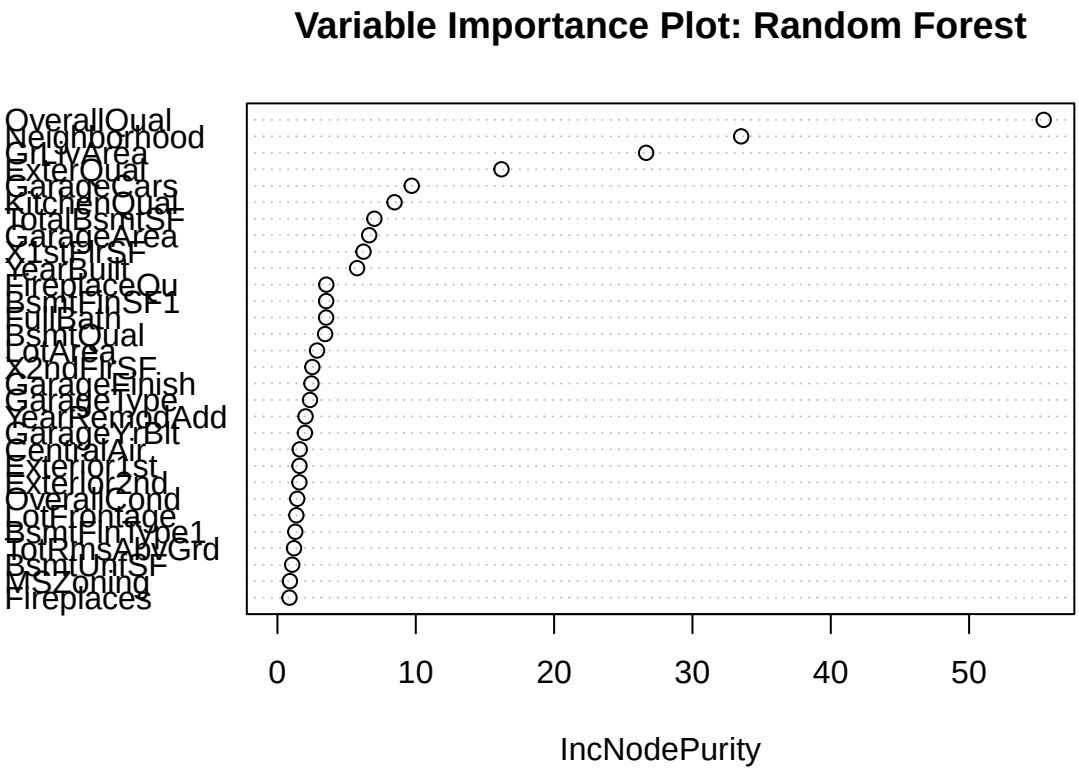
The MSE in for the decision tree is 0.05213, indicating that the house price was predicted relatively accurate. However, we still need more model to tell the performance of the model.

Random Forest

Random forest is an ensemble learning method in machine learning that builds multiple decision trees and combines their results to improve the accuracy and stability of predictions. Unlike decision trees that make a single split at each node, random forest builds multiple decision trees on bootstrapped samples of the original data and randomly selects a subset of features at each split.

Compared to decision trees and LSLR, random forest has several advantages. First, it can handle both numerical and categorical data without requiring data preprocessing, making it a flexible and efficient algorithm. Second, random forest can handle missing data without requiring data imputation (so it can handle the imputed "None" categories well). Third, it reduces overfitting by combining the results of multiple decision trees and using out-of-bag samples to estimate the model's generalization error.

Additionally, random forest is an interpretable algorithm that can provide information on feature importance, making it useful in feature selection and variable importance ranking. In this case, we train our forest model with 100 trees and all data from the house_train. The feature importance plot indicates that the overall quality, neighborhood, and ground living area are the top 3 most important indicator of the house sale price.



The prediciton accuracy of the random forest was higher than the random forest. We derived the mean-squares error of 0.02638 compared with 0.05 MSE of the decision tree, or about 50% of improvement in accuracy.

Conclusion

Based on the analysis of the model output, it can be concluded that overall quality, neighborhood, and ground living area are the top three indicators of house price. Furthermore, it was observed that random forest outperformed decision tree in terms of prediction accuracy. LSLR, on the other hand, proved to be unstable due to the correlation between features.

Therefore, it is recommended to use the random forest algorithm to make predictions on house prices. This algorithm's ability to handle both categorical and numerical data and handle missing data makes it a flexible and efficient algorithm. Additionally, its use of multiple decision trees and the aggregation of their results reduces overfitting, making it a more robust and accurate prediction method. By utilizing random forest, the model's predictive power will be optimized, and more accurate predictions can be made.