Report

Rushabh Khara

2023-05-21

Introduction

The forthcoming report will focus on an extensive statistical analysis and predictive modelling of housing price data from the fictitious city, Heart-Landitopia. Our goal will be to scrutinize two key variables, namely the `price` at which houses are advertised and the presence of a `school` in the vicinity. We aim to unpack the relationship between these variables and a variety of other important covariates, while adequately addressing and discussing inherent uncertainties in the data.

The data under consideration covers diverse parameters such as latitude, longitude, tax rate, presence of homeowners association, HVAC system availability, garage spaces, house view, house age, number of bathrooms, bedrooms, stories, size of the lot and the living space. All of these factors, while being individually unique, contribute collectively towards our two main variables of interest.

In order to derive actionable insights, firstly, we will deploy a minimum of five different classes of models to predict the `price` of a house based on other variables, where our criterion for evaluation will be the `Mean Squared Error`. Secondly, we will implement a similar set of models and algorithms to predict whether or not a school is present nearby, evaluating these models on the `Correct Classification Rate`. Both model selections will be carried out with a strict adherence to justify the predictions made.

Furthermore, we will discuss and compare our models based on several aspects, including uncertainty, predictive rank, and some naive predictions. The best predictive model from these will be examined in light of statistically and scientifically important covariates, along with a discussion on its limitations.

Exploratory Data Analysis (EDA)

EDA will be performed first to understand the data structure, detect outliers and anomalies, uncover underlying patterns, and identify key variables for further analysis, thereby informing our model selection and prediction strategies.

The training data (excluding `id`) consists 5 continuous variables (`price`, `lat`, `lon`, `rate`, `lot`, and `living`), 4 binary categorical variables (`school`, `hoa`, `hvac`, and `view`), 1 ordinal variable (`year`), and 4 discrete variables (`garage`, `bath`, `bed`, and `stories`).

To ensure that R correctly recognizes the categorical variables and presents their respective summaries, it is important to explicitly indicate them using the `factor()` function before invoking the `summary()` function.

Multiple variables in the dataset exhibit missing values, as evident from the following observation. Notably, variables such as `price`, `rate`, `lot`, and `living` demonstrate skewed data distribution, which is evident from the notable difference between their respective mean and median values.

Table 1: NA Table

id	price	school	lat	lon	rate	hoa	hvac	garage	view	year	bath	bed	stories	lot	living
0	0	0	93	89	86	71	94	72	76	77	75	93	64	66	67

Missing data plot in `Appendix` provides insights into the extent and patterns of missing values (`1.3%` Missing), aiding in understanding data completeness and will informs our data pre-processing strategies. An apparent necessity arises to perform data imputation due to the presence of missing values in multiple rows. This poses a challenge during modeling since rows with missing values are typically excluded, leading to data loss and an incomplete representation of the true data relationship within the model.

`Note : Data imputation will be done after EDA.`

Table 2: Summary Table

	price	lat	lon	rate	garage
	Min. :-1.11510	Min. :-1.94049	Min. :-2.87450	Min. :-0.37758	Min. : 0.000
	1st Qu.:-0.44977	1st Qu.:-0.89551	1st Qu.:-0.70092	1st Qu.:-0.31588	1st Qu.: 0.000
	Median :-0.23095	Median :-0.04318	Median : 0.10097	Median :-0.25259	Median : 1.000
	Mean : 0.01709	Mean : 0.01034	Mean :-0.00332	Mean :-0.00848	Mean : 1.312
	3rd Qu.: 0.13940	3rd Qu.: 0.78472	3rd Qu.: 0.71093	3rd Qu.:-0.19267	3rd Qu.: 2.000
	Max. :27.55687	Max. : 2.32792	Max. : 2.43278	Max. : 4.16885	Max. :11.000
	NA	NA's :93	NA's :89	NA's :86	NA's :72

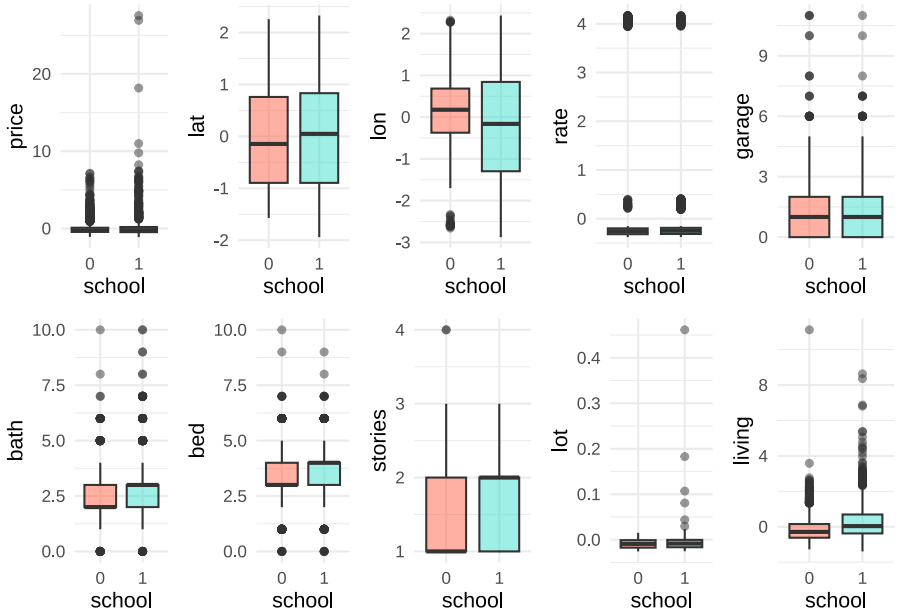
Table 3: Summary Table

	bath	bed	stories	lot	living
	Min. : 0.000	Min. : 0.000	Min. :1.000	Min. :-0.02566	Min. :-1.39103
	1st Qu.: 2.000	1st Qu.: 3.000	1st Qu.:1.000	1st Qu.:-0.01707	1st Qu.:-0.54002
	Median : 3.000	Median : 3.000	Median :1.000	Median :-0.00880	Median :-0.18087
	Mean : 2.683	Mean : 3.444	Mean :1.472	Mean :-0.00862	Mean :-0.00139
	3rd Qu.: 3.000	3rd Qu.: 4.000	3rd Qu.:2.000	3rd Qu.:-0.00070	3rd Qu.: 0.35784
	Max. :10.000	Max. :10.000	Max. :4.000	Max. : 0.46161	Max. :11.11653
	NA's :75	NA's :93	NA's :64	NA's :66	NA's :67

The summary table shows a strange observation with the variable `price` at 27.55687. This anomaly is likely due to an input error, as the value is over a hundred times higher than the mean and median of `price`.

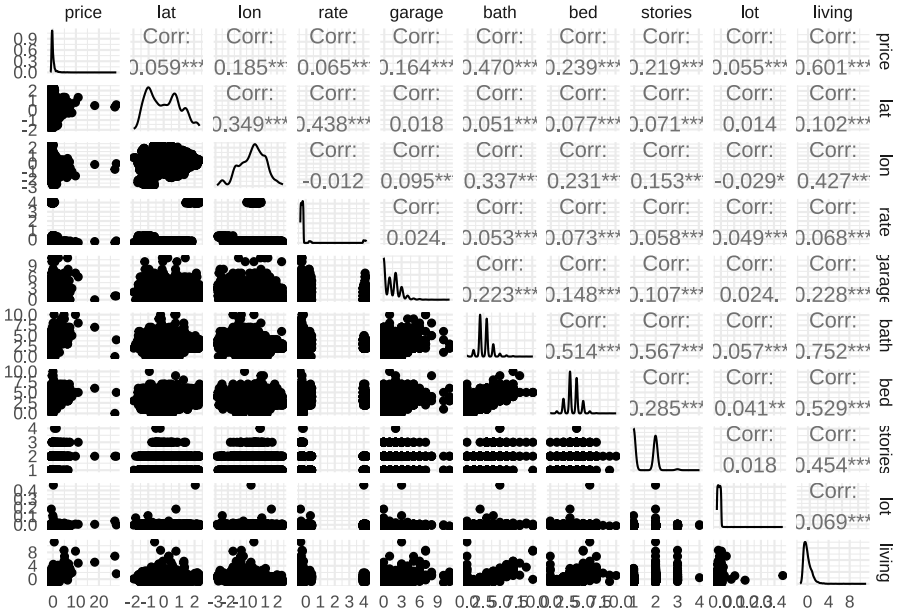
id	price	school	lon	lat	rate	lot	living
1787	27.55687	1	-0.0319511	0.5549837	-0.1547081	-0.0013451	1.607037

Through the examination of boxplots categorized by the variable `school`, key statistical characteristics such as central tendency, spread, and skewness of a variable can be readily discerned. Additionally, the boxplot plots enable the identification of potential outliers in the dataset.



The multivariate plot below exhibits density plots of all the numeric variables and their correlations. Notably, medium correlations are observed between variables such as `bath`, `bed`, and `stories`. This correlation is anticipated, as houses with a greater number of rooms tend to accommodate more individuals, consequently leading to an increased number of bathrooms. Similar reasoning applies to the relationship between `stories` and `bath`. In the present dataset, it is unlikely that multicollinearity will pose a significant issue. Nevertheless, in the event that there is a need to mitigate correlation among the variables, employing transformations could prove beneficial.

`Note: The presented plot does not include rows with missing values, as they are excluded in visualization.`



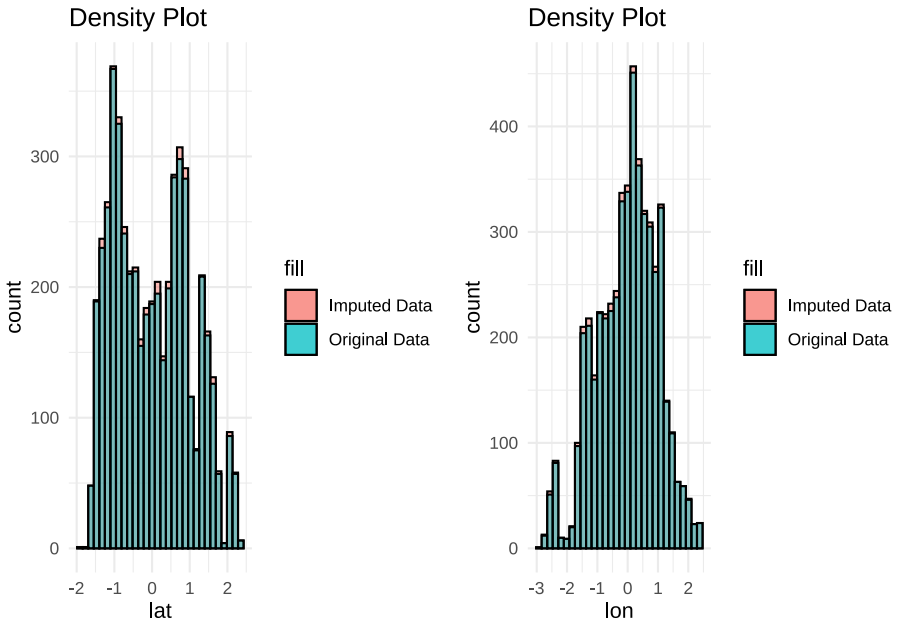
The normality of variables was assessed using the Shapiro-Wilk Test, revealing that none of the variables followed a normal distribution (p-value < 0.05). Checking for normality beforehand is important as violating this assumption can lead to poor model performance. Normality table can be found in Appendix.

Imputation

Imputing missing values before regression modeling is crucial. The `mice()` function is utilized, primarily for multiple imputation. However, specifying `m=1` generates a singular dataset with imputed values, resembling single imputation. This approach has employed data-specific imputation techniques, including `pmm`, `logreg`, and `polr`. With 25 iterations, most variables converge successfully. However, the `view` variable fails to converge, potentially due to high missing values, outliers, intricate patterns, or inappropriate imputation methods for its type.

To assess potential anomalies, we will overlay density plots of the original dataset and the imputed dataset. The comparison reveals that the imputed dataset exhibits a distribution comparable to the original dataset, thereby suggesting a limited introduction of bias through the imputation process.

`Note : Only 2 plots have been shown to save space.`



In order to maintain consistency in the imputation techniques employed, we will apply similar imputation methods to the test dataset.

Transformations

Standardized continuous variables undergo a transformation to achieve a mean of 0 and a standard deviation of 1, rendering them suitable for comparative analysis. As a result, additional transformations for standardized variables may not be necessary or advantageous. Furthermore, categorical and ordinal variables have already been transformed using the `factor()` function, leading to discrete variables. Discrete variables inherently possess a finite range of possible values without a linear or continuous relationship. Consequently, the application of transformations such as standardization or normalization may not be appropriate. Thus, no variable in the dataset requires further transformation. The next step involves commencing the modeling process.

Modelling

A) Predicting price (Continuous Variable)

Our analysis will commence by employing naive prediction models and subsequently progress towards state-of-the-art modeling techniques. To assess the performance of these models in predicting the variable `price`, we will utilize the evaluation metric of `Mean Squared Error`.

`Note: The training data was split into an 80:20 ratio to create a validation dataset before modeling.`

A.1) Multiple Linear Regression

In this study, the feature selection process commenced with the utilization of `stepAIC()` followed by training a multiple linear regression (MLR) model. The mean squared error (MSE) reported herein is a result of k-fold cross-validation followed by validation dataset. By employing k-fold cross-validation alongside a validation dataset, a more comprehensive and robust evaluation of model performance is achieved, promoting generalization.

Table 4: MSE Table

Metric	Value
Multiple Linear Regression MSE (K-fold)	0.5799823
Multiple Linear Regression MSE (Validation)	0.9446783

Based on the regression coefficient summary table, it is apparent that each variable included in the model exhibits statistical significance.

Table 5: Linear Regression Summary

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.2244027	0.0841651	2.6662192	0.0077019
lon	0.0327094	0.0136637	2.3938883	0.0167167
rate	-0.0742678	0.0124354	-5.9723031	0.0000000
hoa1	-0.3424595	0.0316096	-10.8340199	0.0000000
garage	0.0232705	0.0091110	2.5541127	0.0106827
year.L	-0.6346255	0.1898504	-3.3427664	0.0008371
year.Q	0.0785944	0.1710124	0.4595831	0.6458405
year.C	0.1748798	0.1246375	1.4031068	0.1606627
year^4	-0.0528039	0.0783495	-0.6739526	0.5003804
year^5	-0.0598648	0.0451721	-1.3252598	0.1851608
bath	0.2236631	0.0197423	11.3291393	0.0000000
bed	-0.0829159	0.0142287	-5.8273540	0.0000000
stories	-0.1086599	0.0287364	-3.7812648	0.0001583
living	0.7398941	0.0254822	29.0357257	0.0000000

A.2) Ridge Regression

Ridge regression is being used because it offers several benefits that make it well-suited for predicting continuous variables. Specifically, it effectively addresses multicollinearity, balances the bias-variance tradeoff, prevents overfitting, improves robustness to outliers, and provides continuous shrinkage of coefficients. We do not have to make any transformations because we already have standardised predictors.

The first figure below shows our coefficients as a function of lambda. We can observe how the coefficients change with different values of lambda, indicating the impact of regularization on their magnitudes. This provides insights into the extent of shrinkage applied to the coefficients as lambda increases.

In the second plot below, we present the cross-validated MSE as a function of Log(lambda). This plot helps in the selection of an appropriate lambda value when predicting the variable `price`. By examining the trend of the MSE across various lambda values, we can identify the lambda that minimizes the prediction error and achieves a balance between model complexity and generalization. Lambda value of 0.0586 performs the best and is used for predictions MSE below.

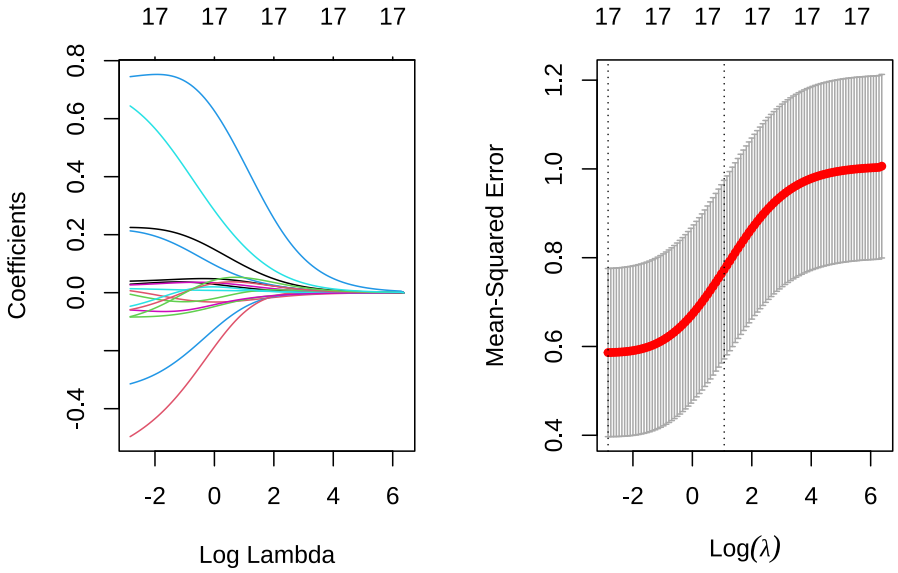


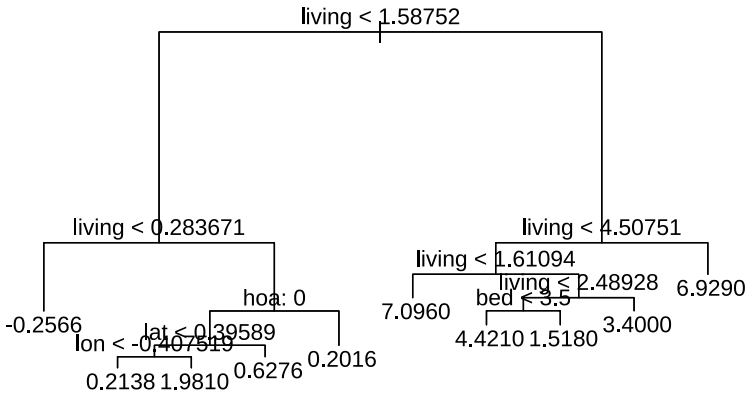
Table 6: MSE Table

Metric	Value
Ridge Regression MSE (K-fold)	0.5736675
Ridge Regression MSE (Validation)	0.9809999

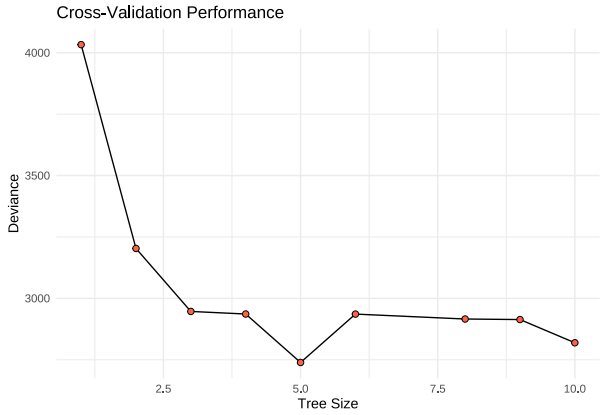
A.3) Decision Tree

Decision trees is being used to predict `price` because they can handle non-linear relationships and partition data into distinct ranges, allowing for effective regression analysis and interpretation.

The decision tree presented below offers a highly interpretable and simplified model that facilitates human understanding. Given its small size, pruning is not necessary at this point, as it does not require substantial computational resources for training.



The Deviance vs. Tree Size plot for decision trees provides us insights into the relationship between the complexity of the tree and the model's deviance. The deviance is a measure of the discrepancy between the predicted values of the model and the actual observed values. It is observed that, best size for our decision tree is 5. This means we prune the tree to size 5.



`Mean Squared Error` from k-fold cross-validation and validation dataset can be seen below.

`Note : Error rates below are produced by pruning tree size to 5.`

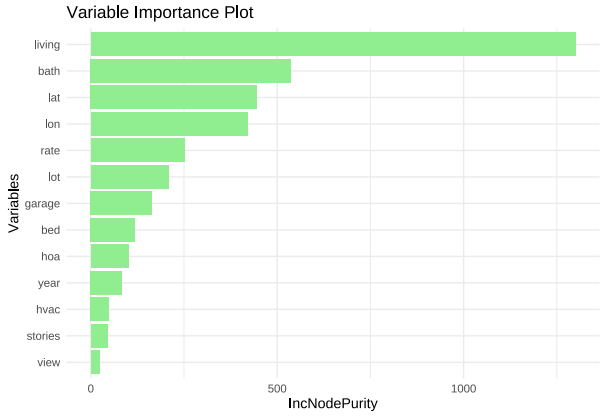
Table 7: MSE Table

Metric	Value
Decision Tree MSE (K-fold)	0.6482248
Decision Tree MSE (Validation)	0.9049313

A.4) Random Forest/Bagging

Bagging is beneficial for predicting continuous variables because it reduces the variance and improves the stability of predictions by combining multiple models trained on bootstrap samples, resulting in more accurate and robust predictions. Random forest is like an upgrade to bagging. Random Forest is utilized for predicting continuous variables due to its ability to capture complex non-linear relationships, handle high-dimensional data effectively, and provide robustness against overfitting and outliers through ensemble averaging. It employs feature subsets to reduce correlation between trees and variance. Our initial Random Forest model includes 1000 trees with an `mtry` value of 4, adhering to the general rule of setting `mtry` as the square root of the total number of predictors. It explains 55.52% of variance with MSE of 0.4472.

Variable importance plot indicates the relative importance of each predictor variable in the model, helping identify influential features in predicting the target variable. This can be also be used as a feature selection method for other regression methods. The `view` variable, being very less important along with `stories`, may have performed poorly due to non-convergence during imputation. `living` and `bath` are the best predictors of `price`.



To determine the optimal model, a k-fold cross-validation technique will be employed. This involves training multiple `randomForest()` models with different configurations and subsequently selecting the model exhibiting the lowest Root Mean Square Error (RMSE). To achieve this, the `trainControl` and `expand.grid` functions will be utilized to train `randomForest()` models with `mtry` of 4, 8, and 13. We have included 13, so we can evaluate `Bagging` alongside and decide which model to

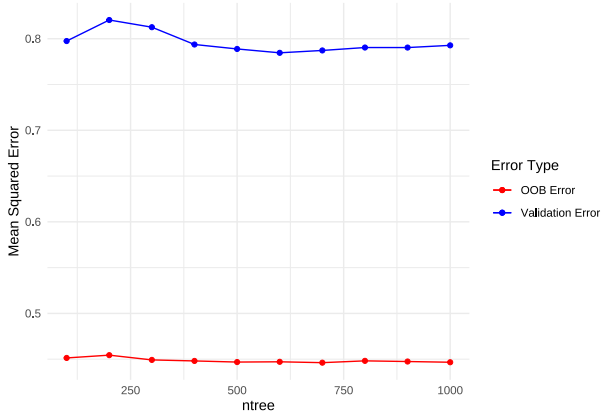
proceed with. The evaluation metrics for all the models are presented in the following table. It is evident that model with `mtry` set to 8 performs the best.

Table 8: Metric Table

mtry	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
4	0.5917163	0.6528751	0.2545018	0.3551610	0.1312170	0.0314535
8	0.5883096	0.6541744	0.2464177	0.3462777	0.1245566	0.0304153
13	0.5995628	0.6420710	0.2481991	0.3386430	0.1183268	0.0294946

The Validation Error vs. OOB Error plot reveals the model's generalization ability and identifies the optimal number of trees for minimizing the MSE. Our analysis indicates that employing 1000 trees yields the lowest MSE and performs the best in terms of both OOB Error and Validation Error. We experimented with tree models exceeding 1000 (up to 5000) in size; however, due to the substantial computational time required, it was impractical to train them on a laptop.

`Note: mtry is set to 8 based on k-fold cross-validation selection.`



This culminates in our ultimate choice of utilizing the `randomForest()` algorithm with 1000 trees and `mtry` set to 8. The Mean Squared Error evaluated through k-fold validation as well as the separate Validation set are demonstrated below.

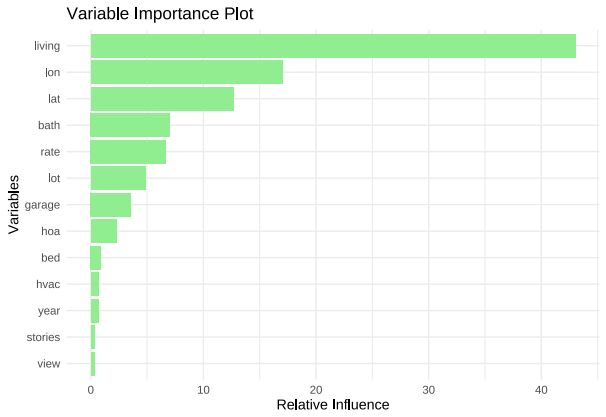
Table 9: MSE Table

Metric	Value
Random Forest MSE (K-fold)	0.3609177
Random Forest MSE (Validation)	0.7845981

A.5) Boosting

Boosting is effective for predicting continuous variables by combining weak models to capture complex relationships. Our initial boosting model consisted of 5000 trees and an interaction depth of 4.

Similar to Random Forest, variable importance plots can be generated to assess the significance of predictor features.



To determine the optimal model, a k-fold cross-validation technique will be employed. This involves training multiple `gbm()` models with different configurations and subsequently selecting the model exhibiting the lowest Root Mean Square Error (RMSE). To achieve this, the `trainControl` and `expand.grid` functions will be utilized to train `gbm()` models with sizes of 1000, 3000, and 5000, while considering interaction depths of 4, 6, and 8. The evaluation metrics for all the models are presented in the following table. It is evident that the model with 1000 trees and an interaction depth of 8 demonstrates superior performance compared to the other models.

Table 10: Metric Table

	shrinkage	interaction.depth	n.minobsinnode	n.trees	RMSE
1	0.01	4	10	1000	0.6465106
4	0.01	6	10	1000	0.6431154
7	0.01	8	10	1000	0.6404879
2	0.01	4	10	3000	0.6694951
5	0.01	6	10	3000	0.6774031
8	0.01	8	10	3000	0.6776216
3	0.01	4	10	5000	0.6782301
6	0.01	6	10	5000	0.6913041
9	0.01	8	10	5000	0.6941249

The plot validates our model's expected behavior, with the MSE consistently decreasing as the number of trees increases. This improvement occurs due to Boosting's iterative learning process, which corrects and enhances the model's performance over time.

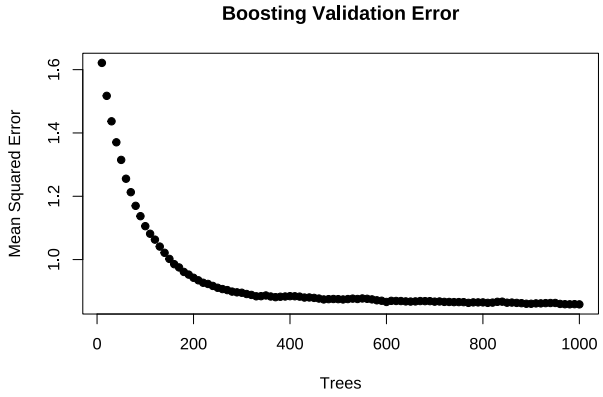


Table 11: MSE Table

Metric	Value
Boosting MSE (K-fold)	0.4102248
Boosting MSE (Validation)	0.7845981

Discussion

Having constructed our models, we will now predict our 50% test data using all the models and analyze the Mean Squared Error (MSE) statistics for K-fold, Validation, and Test. These statistics serve as indicators of how well our models generalize to unseen data. Based on established findings, our random forest models demonstrate superior prediction accuracy when applied to 50% of the test data. In light of this empirical evidence, we will designate the random forest model as our optimal choice for the initial prediction component.

Performing an analysis to assess the fulfillment or violation of assumptions within our model is of utmost importance.

The random forest algorithm is renowned for its considerable flexibility as a modeling technique, which stems from its minimal set of assumptions. It assumes that the utilized sampling methodology is representative of the target population and operates proficiently without explicit distributional assumptions. Consequently, random forests effectively handle various types of data, including skewed and multi-modal distributions, as well as categorical variables, owing to their non-parametric nature.

Moreover, the model demonstrates excellent performance on the training data, which suggests its representativeness of the population and fulfillment of the underlying assumptions.

Table 12: Summary Table

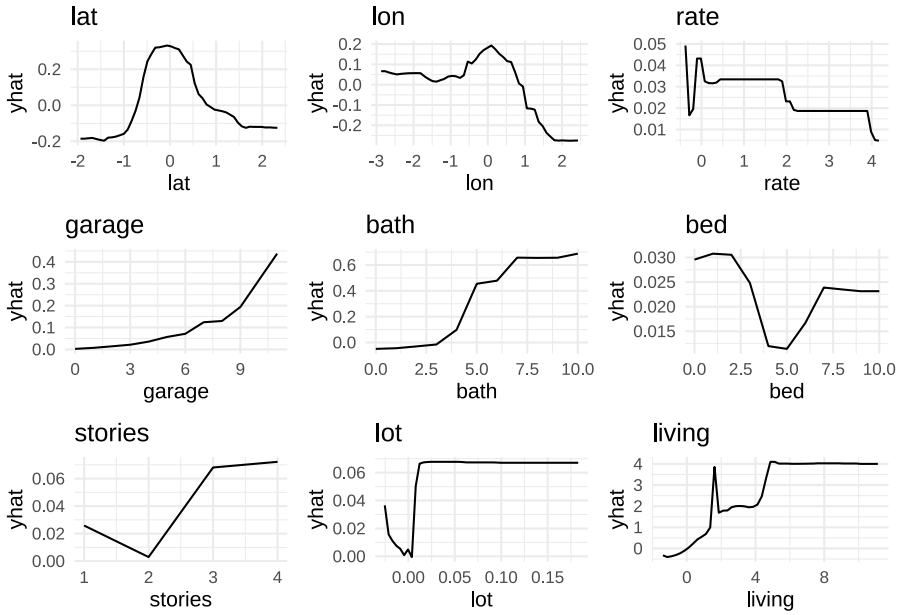
Model	K-fold MSE	Validation MSE	Test MSE
Multiple Linear Regression	0.5799823	0.9446783	0.43900
Ridge Regression	0.5736675	0.9809999	0.44807
Decision Tree	0.6482248	0.9049313	0.63099
Random Forest	0.3609177	0.7845981	0.29468
Boosting	0.4102248	0.7845981	0.31265

Random Forest models are generally considered to be less interpretable compared to simpler models like linear regression. The ensemble nature of Random Forests makes it challenging to interpret the individual decision rules for each tree in the forest. While feature importance measures can provide insights into variable importance, understanding the specific relationships between features and the target variable can be difficult. In the end, it is up to the user to decide the trade-off between performance and interpretability.

Based on assignment specification to prioritize the model with the lowest `Mean Squared Error`, our selection will align with the Random Forest model. By adhering to this suggestion, we aim to minimize the MSE and enhance the accuracy of our predictions.

In Random Forest models, traditional coefficients like those in linear regression are not readily available. However, there are alternative ways to analyze the importance and effects of variables in Random Forest models. Partial Dependence plots below enable us to understand the marginal effect of a single variable on the predicted outcome, independent of the effects of other variables. We have only presented plots for numerical variables to save space.

It can be seen that the partial dependence lines are non-linear which might suggest the presence of thresholds or breakpoints where the effect of the predictor variable changes abruptly. The dependence plots are also indicative of importance of predictor. It can be seen that `living` and `bath` have highest change of magnitude in the predictor variable and hence indicate greater influence in predicting `price`.



B) Classifying school (Categorical Variable)

In the subsequent section, both naive approaches and classification-specific methods will be employed. To assess the performance of these models in predicting the variable `school`, we will utilize the evaluation metric of `Correct Classification Rate`. During the classification process, the selection of a model is based on minimizing the `Classification Error Rate` and maximizing the `Correct Classification Rate`. Therefore, the chosen model is the one that exhibits the highest `Correct Classification Rate`, which is equivalent to the lowest `Classification Error Rate`.

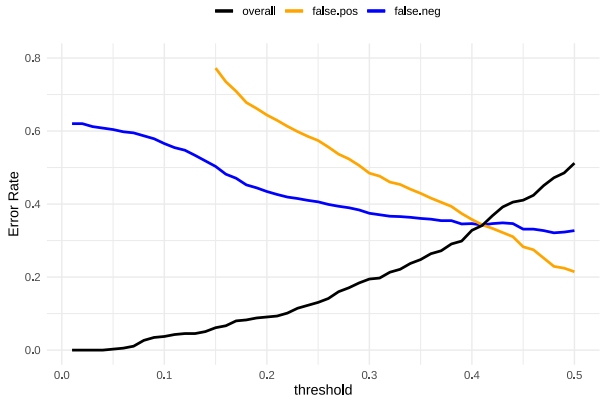
B.1) Logistic Regression

Logistic regression is preferred as our first choice for binary classification due to its interpretability, ability to handle non-linear relationships, provide probability estimates, and perform well with large datasets. Similar to previous prediction of `price`, we will employ `stepAIC()` for feature selection. The best model we get is provided below.

```
Model : school ~ lat + lon + rate + hoa + hvac + view + year + stories + living
```

Adjusting the prediction threshold is crucial prior to computing the optimal estimate for the misclassification rate. The plot below showcases the error rates at different threshold values. Modifying the threshold impacts the balance between misclassification rate, false positive, and false negative rates. Our model achieves the best performance at threshold of `0.41`.

```
Note : Validation dataset has been used to asjust the prediction threshold.
```



Having determined the chosen threshold value, we will employ k-fold cross-validation to estimate the `Classification Error Rate` of our model on training data. Additionally, we have obtained the misclassification rate for the validation data, further evaluating the performance of our model.

Table 13: Classification Error Rate Table

Metric	Value
Logistic Regression Classification Error Rate (K-fold)	0.3252085
Logistic Regression Classification Error Rate (Validation)	0.3424242

The confusion matrix for our logistic regression model is presented below. The model exhibits an `accuracy` of 0.6576, with `prediction confidence` spanning from 0.6271 to 0.6871. Moreover, it achieves a `sensitivity` of 0.6569 and a `specificity` of 0.6587. While our model demonstrates satisfactory performance in classifying the `school` variable, there exist alternative models that warrant further investigation to potentially improve classification accuracy.

We would also not provide confusion matrix in each part to conserve space. Confusion matrix for all following models will be in Appendix.

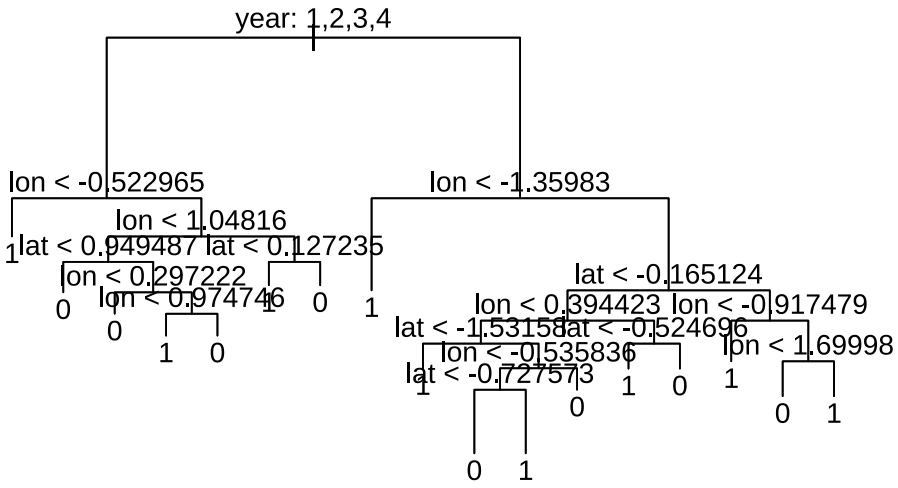
```
Note : Confusion matrix given below is produced using Validation dataset.
```

Table 14: Confusion Matrix

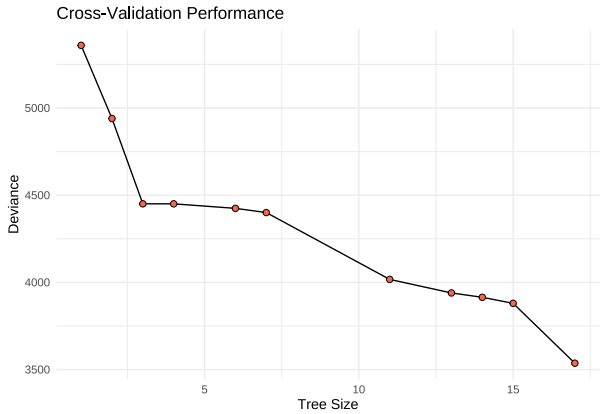
	0	1
0	404	128
1	211	247

B.2) Decision Tree

Decision trees are good for predicting binary categorical variables due to their interpretability, ability to capture non-linear relationships, and feature importance identification. The provided tree offers a simplistic yet interpretable depiction of the decision-making process.



The Deviance vs. Tree Size plot indicates that the optimal tree size for minimizing deviance is determined to be 17. This finding suggests that the tree is already at an ideal size, eliminating the need for further pruning to enhance its performance.



`Classification Error Rate` from k-fold cross-validation and validation dataset can be seen below. The decision tree model demonstrates superior performance compared to logistic regression, offering the advantage of a simpler and more interpretable format.

Table 15: Classification Error Rate Table

Metric	Value
Decision Tree Classification Error Rate (K-fold)	0.1975028
Decision Tree Classification Error Rate (Validation)	0.1979798

B.3) Random Forest/Bagging

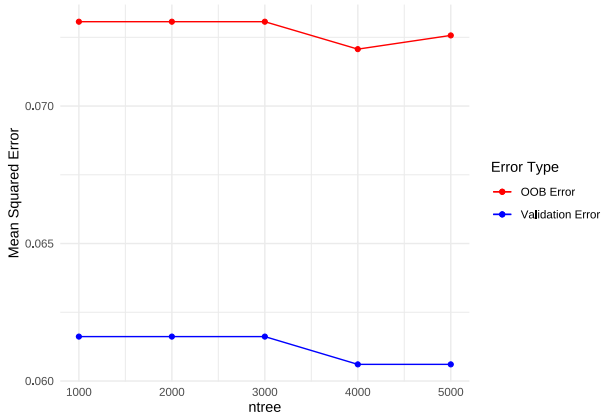
To determine the optimal model, a k-fold cross-validation technique will be employed with folds set to 10. This involves training multiple `randomForest()` models with different configurations and subsequently selecting the model exhibiting the lowest `Classification Error Rate`. To achieve this, the `trainControl` and `expand.grid` functions will be utilized to train `randomForest()` models with `mtry` of 4, 8, and 13. We have included 13, so we can evaluate `Bagging` alongside and decide which model to proceed with. The evaluation metrics for all the models are presented in the following table. It can be seen that bagging technique performs better and hence we will choose 13 as our option for `mtry`.

Table 16: Metric Table

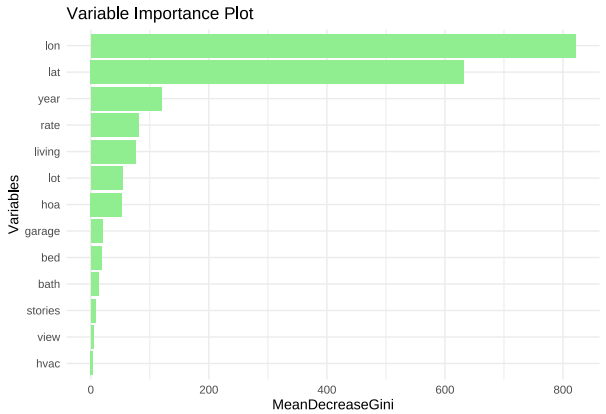
mtry	Accuracy	Kappa	AccuracySD	KappaSD
4	0.8733177	0.7280778	0.0165690	0.0343537
8	0.9117192	0.8121051	0.0109303	0.0234316
13	0.9209431	0.8320560	0.0094420	0.0200185

The Validation Error vs. OOB Error plot reveals the model's generalization ability and identifies the optimal number of trees for minimizing the MSE. Our analysis indicates that employing 4000 trees yields the lowest MSE and performs the best in terms of both OOB Error and Validation Error.

`Note: mtry is set to 13 based on k-fold cross-validation selection.`



Variable importance plot with Gini-index as metric indicates the relative importance of each predictor variable in the model, helping identify influential features in classifying the target variable. This can be also be used as a feature selection method for other regression methods. It can be seen that `lon` and `lat` are really important in predicting class for `school`.



The `Classification Error Rate` evaluated through k-fold validation as well as the separate Validation set are demonstrated below.

Table 17: Classification Error Rate Table

Metric	Value
Random Forest Classification Error Rate (K-fold)	0.0768082
Random Forest Classification Error Rate (Validation)	0.0606061

B.4) Support Vector Machines

SVMs can handle high-dimensional data, non-linearity, outliers, maximize margin, and are suitable for binary classification of categorical variables. The process of finding the optimal SVM model necessitates tuning crucial parameters, namely `cost` and `gamma`. To achieve this, the initial step involves employing the `tune()` function to train various SVM models by systematically varying the cost values of 0.1, 1, 10, 100, and 1000, while simultaneously exploring a range of gamma values spanning from 0.5 to 4. The best tuning parameter for our model are `cost` of 1 with `gamma` set to 0.5 which results in lowest error rate.

Table 18: Best Tuning Parameters Table

cost	gamma	error
1	0.5	0.2234414

The `Classification Error Rate` evaluated through k-fold validation as well as the separate Validation set are demonstrated below.

Table 19: Classification Error Rate Table

Metric	Value
SVM Classification Error Rate (K-fold)	0.2234414
SVM Classification Error Rate (Validation)	0.2353535

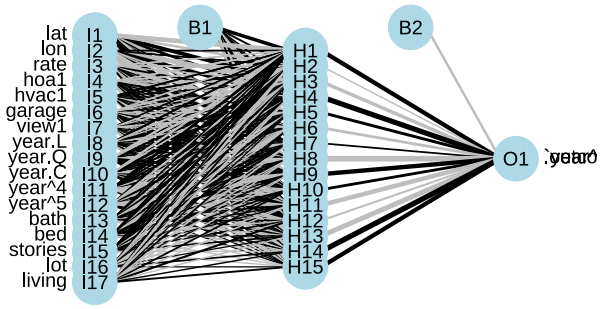
B.5) Neural Network

Neural networks are effective for classifying binary categorical variables due to their ability to capture complex non-linear relationships in data. They can learn intricate patterns, adapt to diverse feature representations, and handle large-scale datasets, making them suitable for modeling and predicting binary categories with high accuracy. However, this doesn't mean that neural networks always provide the best model fit as they are prone to overfitting, noisy or irrelevant features, imbalanced class distributions, and lack of interpretability. We employed k-fold cross-validation with 10 folds to assess the performance of the neural network model. The model was trained using various configurations, including `size` values of 10, 15, and 20, and `decay` values of 0.2, 0.01, and 0.001. The results indicate that the highest accuracy was achieved when using a `size` of 15 and a `decay` of 0.2, as illustrated in the following table.

Table 20: Summary Table

size	decay	Accuracy	Kappa	AccuracySD	KappaSD
10	0.001	0.7982505	0.5663064	0.0207207	0.0468559
10	0.010	0.8040160	0.5808261	0.0307466	0.0634253
10	0.200	0.8172250	0.6082284	0.0267798	0.0587057
15	0.001	0.8149656	0.6036337	0.0244349	0.0522888
15	0.010	0.8114899	0.5973296	0.0284143	0.0624634
15	0.200	0.8264457	0.6288632	0.0156755	0.0359576
20	0.001	0.8154743	0.6076475	0.0237200	0.0512078
20	0.010	0.8214501	0.6186357	0.0223765	0.0506117
20	0.200	0.8264401	0.6281489	0.0215709	0.0445911

The visualization of our optimal neural network model is presented below. However, one limitation of neural networks is their lack of interpretability and informative insights. The increased complexity resulting from larger hidden layers further exacerbates this issue. Additionally, the accuracy achieved by the neural network in this case is not the highest. Consequently, we prioritize the adoption of more interpretable models, even if they exhibit slightly lower accuracy.



The `Classification Error Rate` evaluated through k-fold validation as well as the separate Validation set are demonstrated below.

Table 21: Classification Error Rate Table

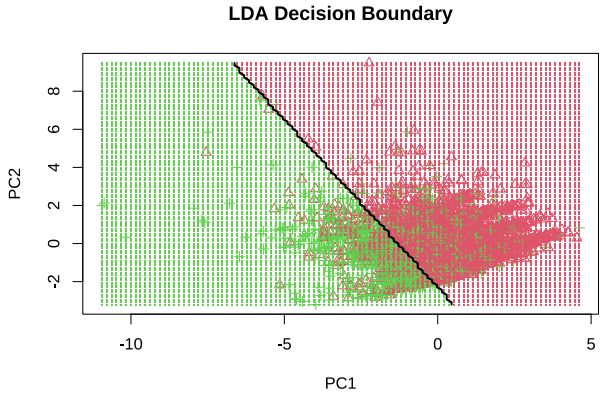
Metric	Value
Neural Network Classification Error Rate (K-fold)	0.1735543
Neural Network Classification Error Rate (Validation)	0.1686869

B.6) Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is good for binary categorical variable classification because it assumes Gaussian distributions, maximizes class separation, and handles multicollinearity well. No variable selection has been taken into account for Linear Discriminant Analysis.

We first reduce our covariate space to two-dimension using Principal Component Analysis (PCA). On applying PCA before LDA, we are reducing the dimensionality of the data to a lower-dimensional space. This reduced space is often chosen to retain most of the variance in the data. The transformed variables in this reduced space are typically uncorrelated. Next, we fit LDA on the reduced data which is further plotted to see the decision boundary between two components. Based on the visual analysis of the plot depicted below, it is apparent that a substantial overlap exists between the two classes. The linear decision boundary, as represented in the plot, fails to adequately capture the underlying relationship between the classes. Consequently, considering the presence of significant overlap, it is recommended to explore non-linear boundaries in order to better capture the complex relationship and improve the classification accuracy.

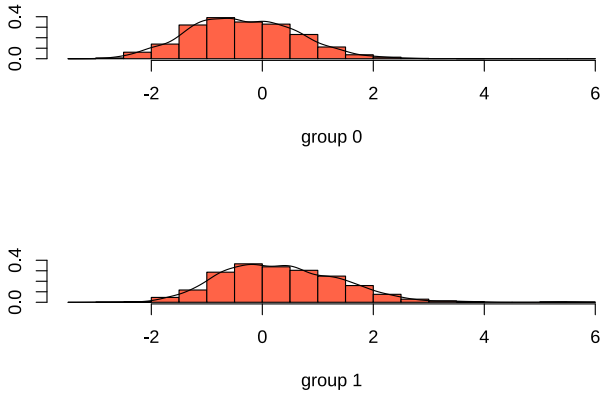
Drawback of dimensionality reduction can also be seen below as plot is not very interpretable and informative in telling us the decision making process.



We can see the 95% confidence interval of decision boundary below. This is based on 1000 bootstrap samples.

Lower.Bound	Upper.Bound
-0.590273	-0.5382574

The presented plot depicts the values of linear discriminant functions for groups 0 and 1. The histograms of the distributions look normal after performing PCA before LDA.



The `Classification Error Rate` evaluated through k-fold validation as well as the separate Validation set are demonstrated below. We receive an significant error rate of which is very high as compared to other models.The interpretability of the LDA model is challenging, and its performance does not justify the computational resources required in this case. As a result, LDA may not be the preferred choice for prediction and comparison with other models. Alternative approaches should be considered to overcome these limitations and improve the overall predictive performance.

Table 22: Classification Error Rate Table

Metric	Value
LDA Classification Error Rate (K-fold)	0.3391521

Discussion

Based on established findings, our random forest models demonstrate superior prediction accuracy when applied to 50% of the test data. In light of this empirical evidence, we will designate the Bagging model as our optimal choice for the second prediction component.

Table 23: Summary Table

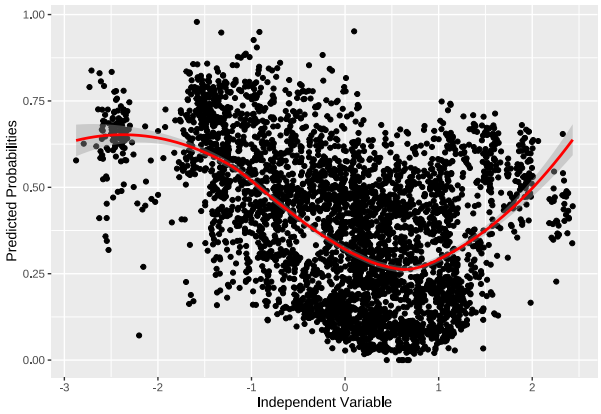
Model	K-fold Classification Error	Validation Classification Error	Test Correct Classification Rate
Logistic Regression	0.3252085	0.3424242	0.65866
Decision Tree	0.1975028	0.1979798	0.77466
Bagging	0.0768082	0.0606061	0.92066
SVM	0.2234414	0.2353535	0.76133
Neural Network	0.1735543	0.1686869	0.81333

In consideration of our prior analysis of the random forest model in the preceding section which is highly similar to bagging, it is advantageous to explore an alternative model, even if it may not exhibit optimal performance.

Logistic regression would be the ideal choice to interpret the relation between predictors and response variable. It provides estimates of the relationship in terms of odds ratios, allowing for the interpretation of the impact of each predictor on the probability of the response occurring, making it a valuable tool for studying the association between predictors and the response in a binary classification context. Logistic regression offers several advantages for analytical purposes, which will be further explored in subsequent discussions.

Performing an analysis to assess the fulfillment or violation of assumptions within our model is of utmost importance. Logistic regression assumptions include: a binary outcome, independence of observations, linearity in logit, no multicollinearity, and a sufficient sample size. These assumptions are crucial for reliable and valid interpretations of logistic regression results.

First assumption for our model is satisfied as `school` is a binary categorical variable. It can be assumed that data provided to us is independent. In the context of assessing the linearity assumption in logit, a visual examination of the Predicted Probabilities vs. Independent Variable plot reveals a clear non-linear relationship which shows violation of assumption.



Multicollinearity can be evaluated by examining the VIF (Variance Inflation Factor) values, which measure the inflation of the estimated regression coefficient's variance resulting from multicollinearity. In this analysis, the VIF values for the `year` variable are exceptionally high, suggesting the presence of multicollinearity.

Table 24: VIF Table

	x
lat	8.563102e+00
lon	8.019665e+00
rate	6.966266e+00
hoa1	8.106519e+00
hvac1	5.475112e+00
view1	5.224088e+00
year.L	4.463336e+06
year.Q	9.844342e+06
year.C	4.341341e+06
year^4	1.068144e+06
year^5	1.140355e+05
stories	7.135726e+00
living	1.084455e+01

Finally, with a sample size of 4010 after excluding 990 observations for the validation dataset, the number of remaining observations is considered sufficiently large. Hence, it can be assumed that the sample size meets the requirements of the assumption.

Table 25: Linear Regression Summary

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-3.0766051	36.3003531	-0.0847541	0.9324569
lat	0.3603374	0.0464438	7.7585660	0.0000000
lon	-0.1802532	0.0449364	-4.0112915	0.0000604
rate	-0.3564919	0.0428144	-8.3264454	0.0000000
hoa1	0.6875240	0.0901376	7.6274911	0.0000000
hvac1	0.1864153	0.0993482	1.8763845	0.0606025
view1	0.1717332	0.0877157	1.9578388	0.0502489
year.L	8.8284557	130.1603854	0.0678275	0.9459229
year.Q	-5.2671200	118.8195108	-0.0443287	0.9646424
year.C	3.3452852	81.1697562	0.0412134	0.9671257
year^4	-1.7814087	41.1617237	-0.0432783	0.9654797
year^5	0.5159554	13.7220063	0.0376006	0.9700061
stories	-0.5152252	0.0806426	-6.3889957	0.0000000
living	0.3456873	0.0658491	5.2496915	0.0000002

The summary table above gives us more insight into importance of predictors in predicting response. As expected `year` is not significant due to multicollinearity. This might be due to `stepAIC()` including the non-significant variable as removing it increases the AIC value. This shows that we should be careful and not entirely be dependent on `stepAIC()` for model selection.

This concludes our analysis of second predicting component of the report.

Conclusion

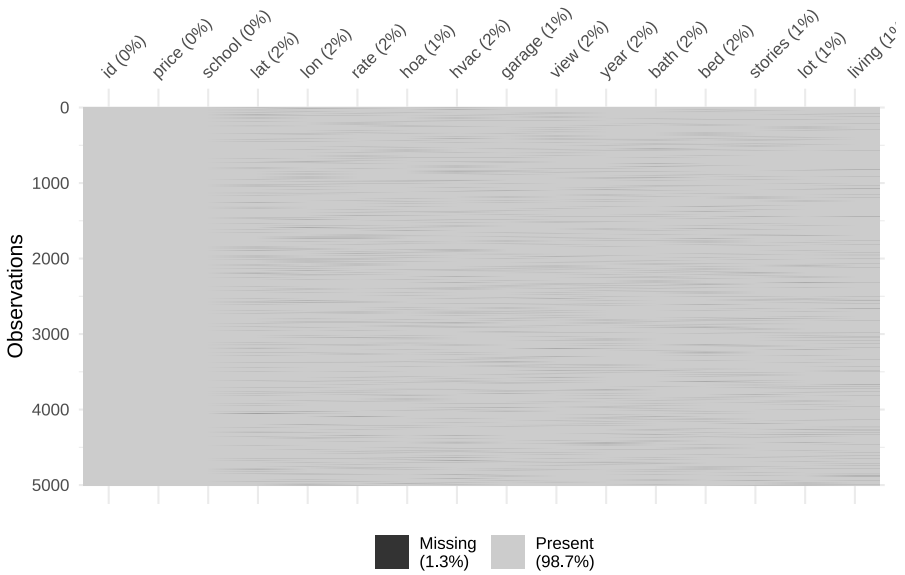
In the first prediction part, the random forest model was employed to analyze the relationship between various predictors and the target variable, `price`. The statistical analysis revealed that `living` and `bath` were the most significant indicators of `price`. Larger living spaces were found to provide greater flexibility and perceived value, while multiple bathrooms were associated with enhanced convenience and a higher quality of life. Furthermore, these variables also reflected the property's capacity to accommodate a larger number of occupants. Additionally, the location of the property emerged as a crucial factor, as prices tended to be higher in well-developed areas with more facilities.

In the second classification part, the analysis identified `lon` and `lon` as the most important predictors of the target variable, `school`. The significance of location for schools stems from its impact on student accessibility. Proximity to residential areas ensures convenience, minimizes transportation challenges, and attracts families to the institution.

The alignment between the statistical findings and the scientific intuition validates the efficacy of the modeling approach in identifying the most influential variables for predicting the response variable.

Appendix

1) Missing Value Plot for Training Data



2) Normality Test for Variables

Table 26: Normality Summary

vars	statistic	p_value	sample
price	0.5162528	0	4010
lat	0.9576608	0	4010
lon	0.9892063	0	4010
rate	0.3163203	0	4010
garage	0.8277296	0	4010
bath	0.8633080	0	4010
bed	0.9127550	0	4010
stories	0.6700385	0	4010
lot	0.8788493	0	4010
living	0.8340071	0	4010

3) Confusion Matrix for Classification Models

Table 27: Confusion Matrix Decision Tree

	0	1
0	571	152
1	44	223

Table 28: Confusion Matrix Bagging

	0	1
0	590	35
1	25	340

Table 29: Confusion Matrix SVM

	0	1
0	531	149
1	84	226

Table 30: Confusion Matrix Neural Network

	0	1
0	546	98
1	69	277