SCHOOL OF GEOGRAPHY **UNIVERSITY OF LEEDS**	UNIVERSITY OF LEEDS

COURSEWORK COVERSHEET

Student ID number	2	0	1	6	6	9	1	3	8
Module code	GEOG5917M								
Module title	Big Data and Consumer Analytics								
Assignment title	Extreme Gradient Boosting Linear Model For House Price Prediction With The Boston House Dataset								
Marker									
Declared word count	2440								

By submitting the work to which this sheet is attached you confirm your compliance with the University's definition of Academic Integrity as: "a commitment to good study practices and shared values which ensures that my work is a true expression of my own understanding and ideas, giving credit to others where their work contributes to mine". Double-check that your referencing and use of quotations is consistent with this commitment.

You also confirm that your declared word count accurately reflects the number of words in your submission, excluding the overall title, bibliography/reference list, text/numbers in tables and figures (although table and figure captions are included in the word count).

EXTREME GRADIENT BOOSTING LINEAR MODEL FOR HOUSE PRICE PREDICTION WITH THE BOSTON HOUSE DATASET

INTRODUCTION

The difficulty of projecting property prices has long been a source of concern in the real estate industry. Predicting property prices accurately is crucial for both buyers and sellers to make sound choices. Machine learning models are one of the most effective ways to solve this challenge (Jiang et al., 2022). The Extreme Gradient Boosting Linear (xgbLinear) model has garnered a great deal of interest among machine learning algorithms because of its capacity to handle huge amounts of data and its quick and accurate prediction. XgbLinear is an extension of XGBoost. XgbLinear's base learner is linear regression, which makes it more interpretable and less prone to overfitting (Chen and Guestrin, 2016).

This project intends to construct a predictive xgbLinear model for house prices using the Boston House dataset. The Boston House dataset is a popular benchmark dataset for regression research. Another objective of this project is to assess how well the xgbLinear model predicts house prices using this dataset and use the feature importance analysis to determine the most relevant house price predictors. The final model can be used by real estate agents in Boston to predict if their investment will yield profit or result in a loss.

Scalability, faster learning speeds, in all situations, and out-of-core processing to let data scientists analyse billions of instances on a single desktop are the advantages of the XGBoost framework (Chen and Guestrin, 2016). It is very flexible, efficient, portable, and supervised. It has the advantage of being able to handle missing values. It employs sophisticated regularisation (L1 and L2) to improve model generalisation issues (Dalal et al., 2022).

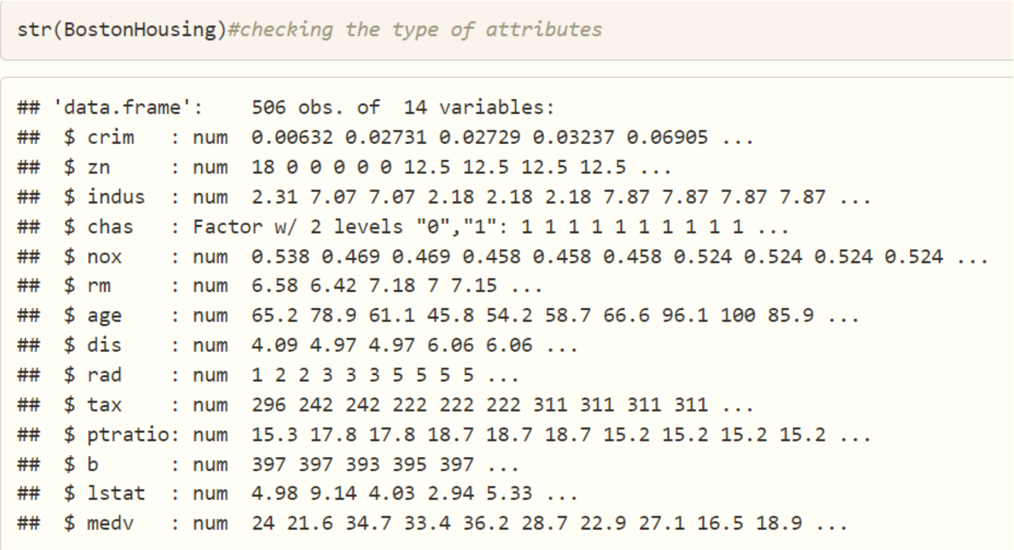
METHODS

The Boston housing dataset was taken from the ‘mlbench’ package in R (Leisch et al., 2009). It includes data on numerous elements that may influence housing costs, such as the average number of rooms, the distance to employment centers, and the crime rate. The dataset contains 506 records and 14 variables among which ‘MEDV’ is the target variable and the rest are the predictors. Table 1 describes all the variables (Harrison and Rubinfeld, 1978). The objective is to create a model that can accurately predict the median price of privately owned properties.

Variable	Description
crim	per capita crime rate by town
zn	proportion of residential land zoned for lots over 25,000 sq.ft
indus	proportion of non-retail business acres per town
chas	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
nox	nitric oxides concentration (parts per 10 million)
rm	average number of rooms per dwelling
age	proportion of owner-occupied units built prior to 1940
dis	weighted distances to five Boston employment centres
rad	index of accessibility to radial highways
tax	full-value property-tax rate per USD 10,000
ptratio	pupil-teacher ratio by town
b	1000(*B*–0.63)2 where *B* is the proportion of blacks by town
lstat	percentage of lower status of the population
medv	Median value of owner-occupied homes in $1000’s

Table 1: Description of variables

The data type of the variables and their examples can be seen in Figure 1.

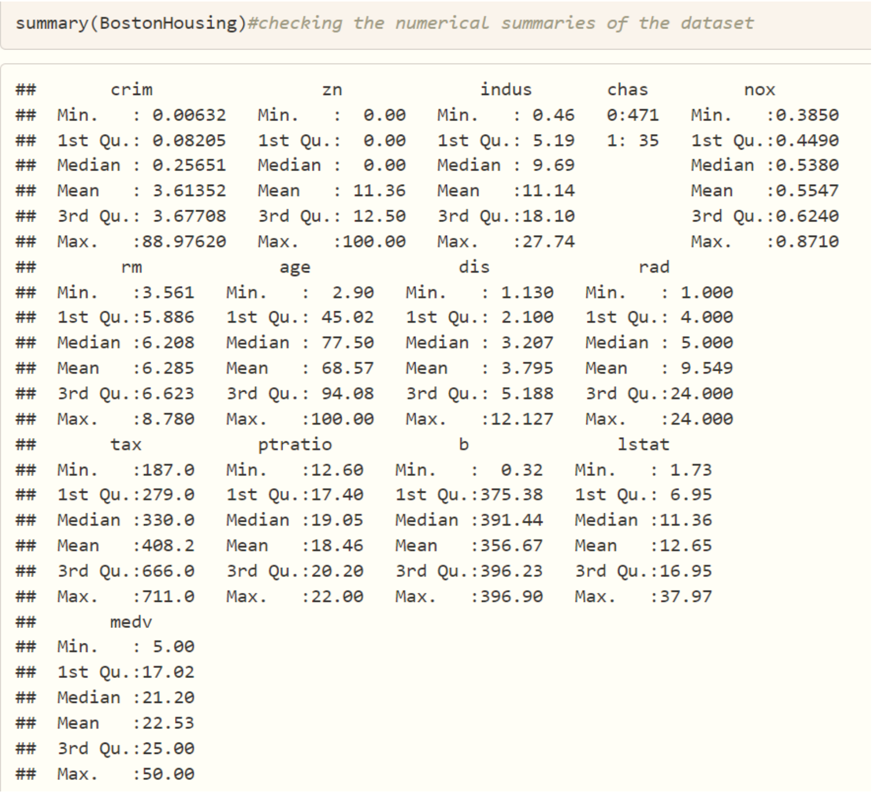


```
str(BostonHousing)#checking the type of attributes
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : num  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ b      : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

Figure 1: Datatype of variables with examples

The numerical summary of the dataset is shown in Figure 2. ‘chas’ is a categorical variable with 2 levels ‘0’ and’1’, and the rest are numerical variables.

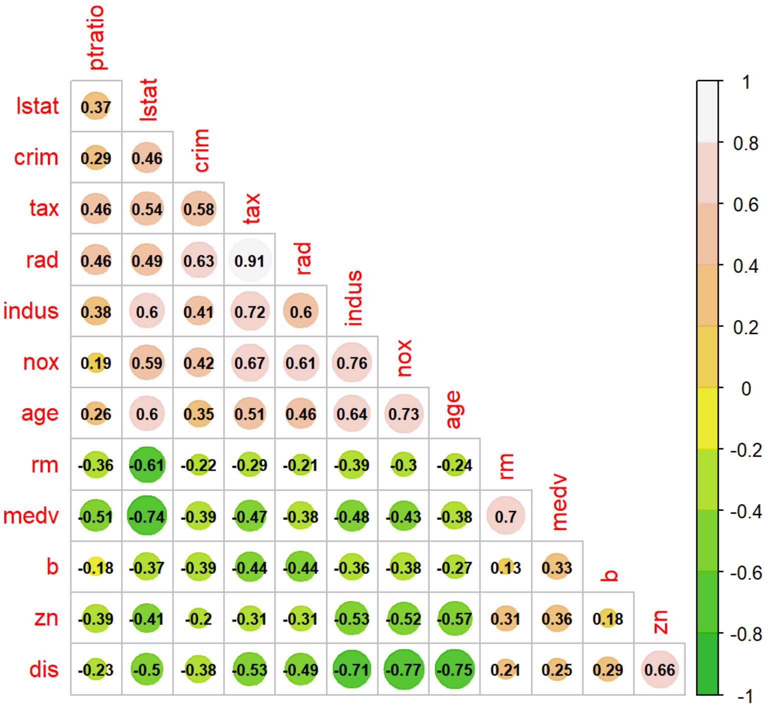


```
summary(BostonHousing)#checking the numerical summaries of the dataset
```

```
##       crim                zn             indus       chas         nox
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   0:471   Min.   :0.3850
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1: 35   1st Qu.:0.4490
##  Median : 0.25651   Median :  0.00   Median : 9.69           Median :0.5380
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14           Mean   :0.5547
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10           3rd Qu.:0.6240
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74           Max.   :0.8710
##        rm             age              dis              rad
##  Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000
##  1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100   1st Qu.: 4.000
##  Median :6.208   Median : 77.50   Median : 3.207   Median : 5.000
##  Mean   :6.285   Mean   : 68.57   Mean   : 3.795   Mean   : 9.549
##  3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188   3rd Qu.:24.000
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000
##       tax           ptratio            b              lstat
##  Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73
##  1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95
##  Median :330.0   Median :19.05   Median :391.44   Median :11.36
##  Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95
##  Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97
##       medv
##  Min.   : 5.00
##  1st Qu.:17.02
##  Median :21.20
##  Mean   :22.53
##  3rd Qu.:25.00
##  Max.   :50.00
```

Figure 2: Numerical summary of the dataset

A thorough analysis of the dataset found no duplicate records as well as no missing records. The correlation between the variables was found using the Pearson correlation matrix as shown in Figure 3.



	ptratio	lstat	crim	tax	rad	indus	nox	age	rm	medv	b	zn
lstat	0.37											
crim	0.29	0.46										
tax	0.46	0.54	0.58									
rad	0.46	0.49	0.63	0.91								
indus	0.38	0.6	0.41	0.72	0.6							
nox	0.19	0.59	0.42	0.67	0.61	0.76						
age	0.26	0.6	0.35	0.51	0.46	0.64	0.73					
rm	-0.36	-0.61	-0.22	-0.29	-0.21	-0.39	-0.3	-0.24				
medv	-0.51	-0.74	-0.39	-0.47	-0.38	-0.48	-0.43	-0.38	0.7			
b	-0.18	-0.37	-0.39	-0.44	-0.44	-0.36	-0.38	-0.27	0.13	0.33		
zn	-0.39	-0.41	-0.2	-0.31	-0.31	-0.53	-0.52	-0.57	0.31	0.36	0.18	
dis	-0.23	-0.5	-0.38	-0.53	-0.49	-0.71	-0.77	-0.75	0.21	0.25	0.29	0.66

Figure 3: Correlation matrix

As price is proportional to the number of rooms, the variable 'rm' has a high positive correlation with the target variable 'medv'. 'lstat' and 'pratio', on the other hand, show a strong negative correlation with 'medv,' since lower housing prices indicate lower financial status and, hence, less education among the population.

The outliers in the data were analysed using boxplots in Figure 4.

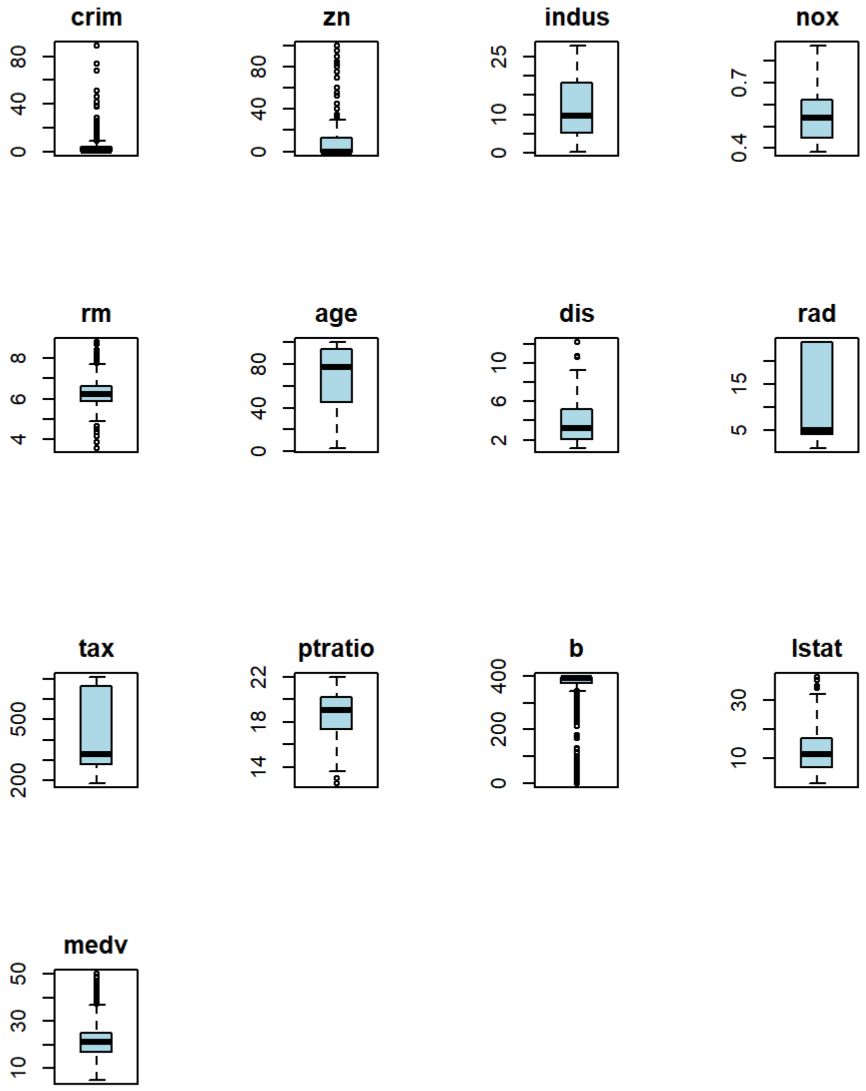
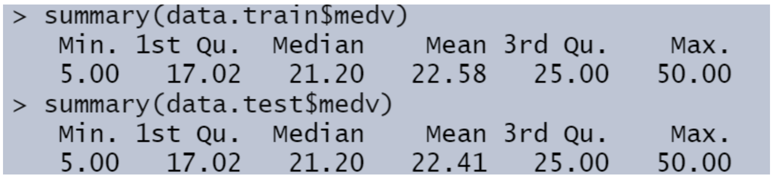


Figure 4: Boxplots shownig the outliers in variables

Variables ‘zn’, ’b’, and ’crim’ have sufficient outliers, which could potentially affect the prediction model. However, since these variables have a sufficient correlation with the target variable, there were used for modeling.

For the supervised learning algorithm, the data needs to be split into two, one for model creation and the other for validation (Comber and Brunsdon, 2020). Hence, the data was split into train and test data using 70:30 splitting ratio randomly, ensuring the same distribution for both as shown in Figure 5. The training set was employed to fit and tune the model and the test set was used to ensure that the model performs predictions without any bias on unknown data (Xenochristou and Kapelan, 2020).



```
> summary(data.train$medv)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   17.02   21.20   22.58   25.00   50.00
> summary(data.test$medv)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.00   17.02   21.20   22.41   25.00   50.00
```

Figure 5: Summary of test and train data

The variables such as ‘tax’, ‘b’, ‘nox’, etc., had different units of measurement ranging from 0.4 to 500. To minimise scale disparities between various units of measurement while fitting the model, input data, except the target variable ‘medv’, was rescaled (normalised). Scaling was done using the conventional method of employing z-scores, to prevent the variables having higher values from dominating the model fitting. Z-scores modify all the data irrespective of their units, to acquire a mean value of zero and a standard deviation of one, without affecting the distribution of the data as in equation (1).

$$\mathrm{z} = \frac{x-\bar{x}}{\sigma_x} \qquad (1)$$

,where $\bar{x}$ is the mean of x and σ_x is the standard deviation of x (Comber and Brunsdon, 2020).

To ensure model reliability, models must be assessed throughout creation using k-fold cross-validation or leave-one-out cross-validation. These procedures include resampling, which divides the data into k subsets or 'folds'. Each fold is used as a hold-out set once. The model is trained using the remaining k-1 folds and tested using the hold-out set to obtain the evaluation score. This process is repeated k times and the average of the retained evaluation scores provides an overall estimate of fit. For this project, k=10 was chosen, which splits the dataset by 10 folds (Comber and Brunsdon, 2020).

RESULTS

The model building was performed with the ‘MAE’ metric using the ‘xgbLinear’ method in the R caret package, which implements the XGBoost algorithm. XGBoost is a boosting method that integrates many non-ensemble Distributed Matrix Multiplication approaches into a single classifier (Chen and Guestrin, 2016; Bertolini, 2021). Each basic classifier in this technique is trained before being successively added to the model to correct misclassifications at earlier levels by changing the weights of various features (Breiman, 1996; Bertolini,2021). This process is repeated continuously until the discrepancy between the predicted and actual values of the dependent variable is as minimal as possible (Friedman, 2002; Bertolini, 2021). The 'xgbLinear' method employs a linear penalty function that limits the weights of the observations at each iteration. xgbLinear has an architecture that is an ensemble of Generalized Linear Model algorithms. It uses a gradient descent algorithm (Bertolini, 2021). The properties of the xgbLinear model include scalability, linearity, regularisation, feature selection, and interpretability (Chen and Guestrin, 2016).

The model parameters specified before training are referred to as hyperparameters. It refers to the specification of a collection of input parameters that impact the model structure and, as a result, the outputs (Xenochristou and Kapelan, 2020). There are four hyperparameters in this model (Bertolini, 2021; Pesantez-Narvaez et al., 2019):

- Number of iterations (nrounds) defines the number of iterations for boosting.
- Learning rate (η) defines the step size shrinkage to prevent overfitting.
- L1 Regularization Parameter (α) penalises the sum of absolute value of the weights.
- L2 Regularization Parameter (λ) penalises the sum of the square weights.

The best combination of hyperparameters helps the model to perform well and these can be found by evaluating the model's performance. A general strategy adopted is to:

1. Split the data into train ($S = \mathrm{X}, y$) and test set ($S' = \mathrm{X}', y'$).
2. Using S and a given set of tuning parameters, calibrates a function f.
3. Use X' to find $f(\mathrm{X}')$.
4. The final results are compared to y'.

The measures of fit used for comparison are listed below:

- Root Mean Square Error (RMSE) = $\sqrt{\sum \frac{\left(y' - f(x')\right)^2}{n}}$
- The coefficient of determination, $R^2 = 1 - \frac{\Sigma\left(y' - f(x')\right)^2}{\Sigma\left(y' - \bar{y'}\right)^2}$
- Mean Absolute Error (MAE) $= \frac{\Sigma|y' - f(x')|}{n}$

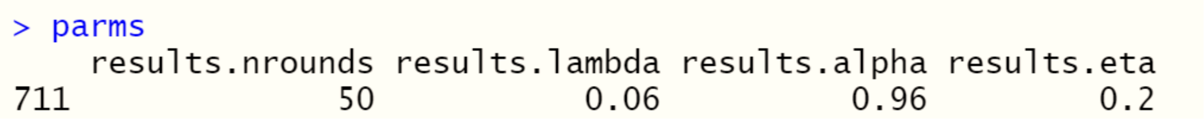
For RMSE and MAE, a smaller value indicates better prediction, whereas, higher values stand for better results. MAE is the most commonly used measure of fit in Machine Learning models (Comber and Brunsdon, 2020).

In this model, a random grid search was employed to find the best tuning parameters. The tuning grid used to pass these hyperparameters to the training model to efficiently tune the model is shown in Table 2.

Parameter	Values
nrounds	50
η	[0.2, 0.25, 0.3, 0.35, 0.4]
α	[0.8, 0.81, 0.82,…,1]
λ	[0, 0.01, 0.02,…,0.2]

Table 2: Tuning grid used for training

The best tuning parameters obtained from grid search is shown in Figure 6.



```
> parms
    results.nrounds results.lambda results.alpha results.eta
711              50           0.06          0.96         0.2
```

Figure 6: Hyperparameters for best tuned model

The predicted over observed values of house prices after tuning from the xgbLinear model are shown in Figure 7.

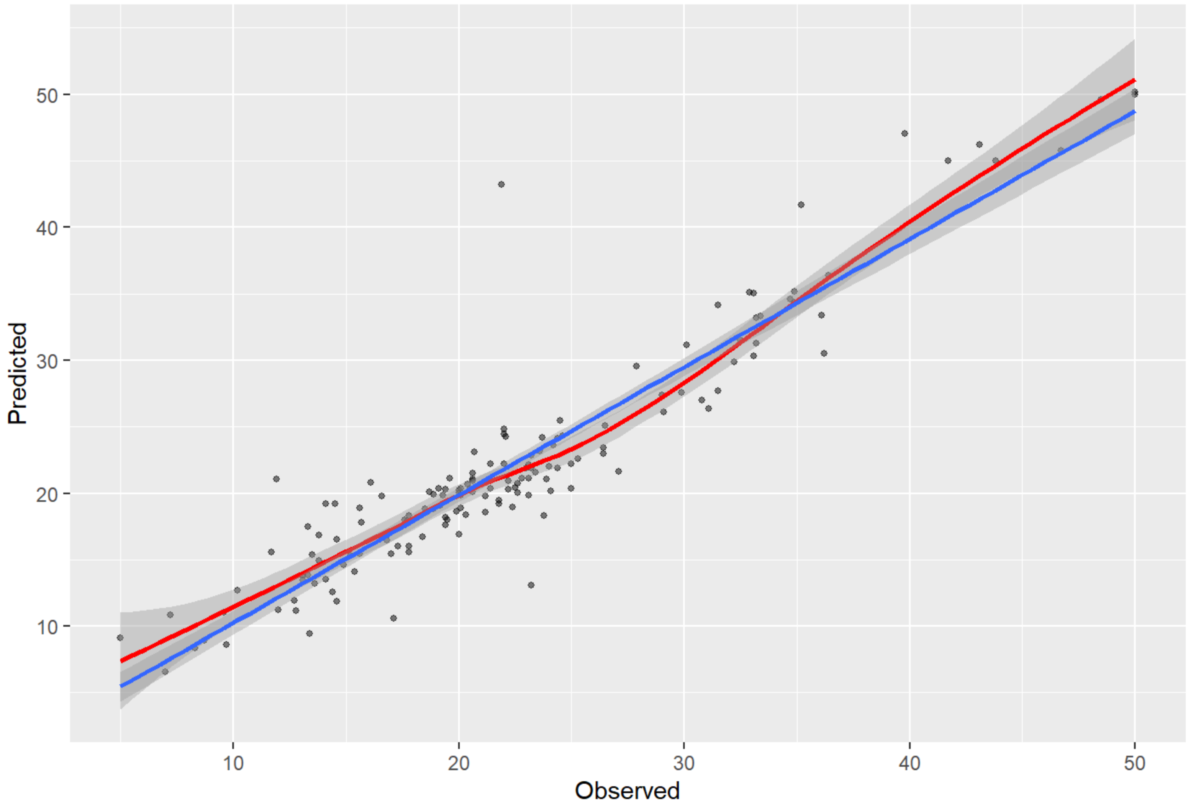
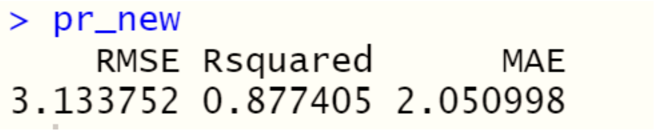


Figure 7: Predicted v/s observed values house price after tuning

The blue line indicates the trend line of model fit and the red line indicates the loess trend, which is the variation of the observed from prediction. It can be seen that most of the points in the lower half and some portion of the upper quarter lie close to the prediction line and the loess trend line is parallel to the prediction line in those regions, which shows that the model does well in predicting prices in the ranges $10,000-$23,0000 and $33,000-$37,000. However, some points lie below and above the trend line, which shows that the model overestimated the former and underestimated the latter. The measures of fit showing the overall model performance are shown in Figure 8. The low RMSE and MAE values and a high value close to 1, obtained for this model confirm that the xgbLinear model for Boston house price prediction performs well.



```
> pr_new
    RMSE Rsquared      MAE
3.133752 0.877405 2.050998
```

Figure 8: Measures of fit for the house price prediction model

The variable importance of the model was checked and the top 10 variables are shown in Figure 9. It shows that the lower status of population, number of rooms, and weighted distances to 5 employment centres have a significant role in the prediction.

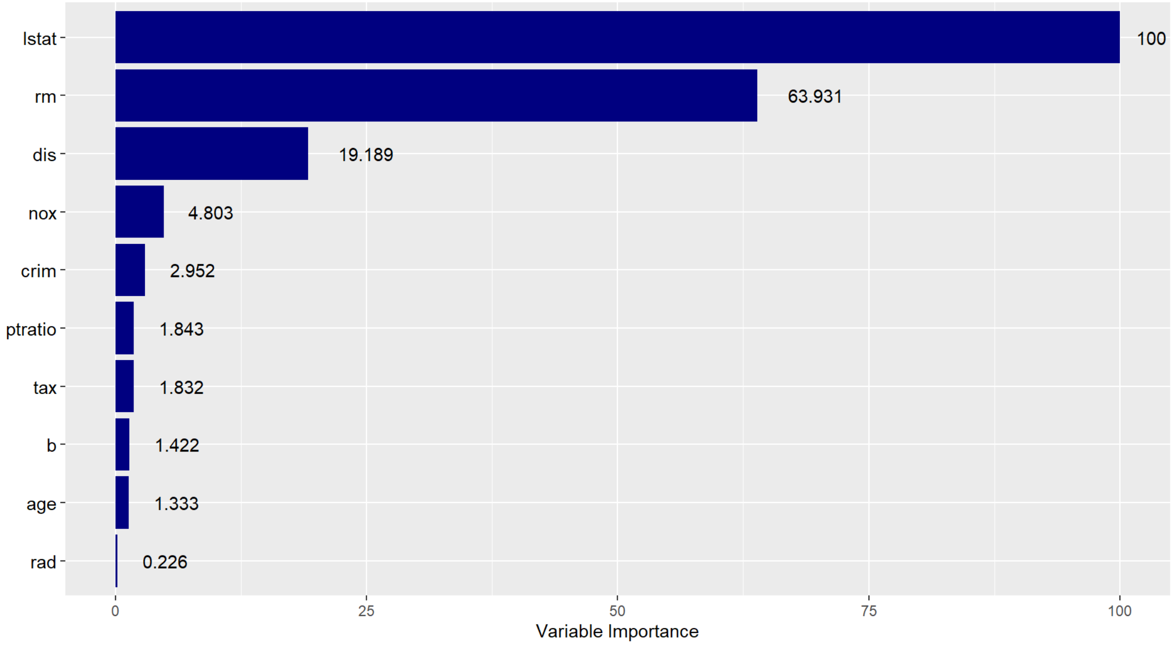


Figure 9: Top 10 important features in the model

This model can be used by real estate agents to predict the prices of houses with the same attributes in a different locality, thereby ensuring the safety of the investment. Homeowners who are confused about selling their homes can use this model to predict if the prices would go up in the future.

DISCUSSION

An initial model was created using the training data, the xgbLinear method, and the train control function in the caret package of R, which performs ten iterations of cross-validation for in-model sampling. The model obtained before tuning is shown in Figure 9. Those points were mostly away from the trendline. The measures of fit for the model were analysed and are shown in Figure 10. Using the obtained model, the ranges of values for hyperparameters were analysed and found. Used those values to make the tuning grid and trained the final model. The best hyperparameters used for the model were found and shown in Figure 6. The resulting measures of fit (Figure 8) showed improved results as the RMSE and MAE values decreased and the value increased. Also, most of the points got closer to the prediction trend line in the final model compared to the initial model. Thus, the final model got improved performance.

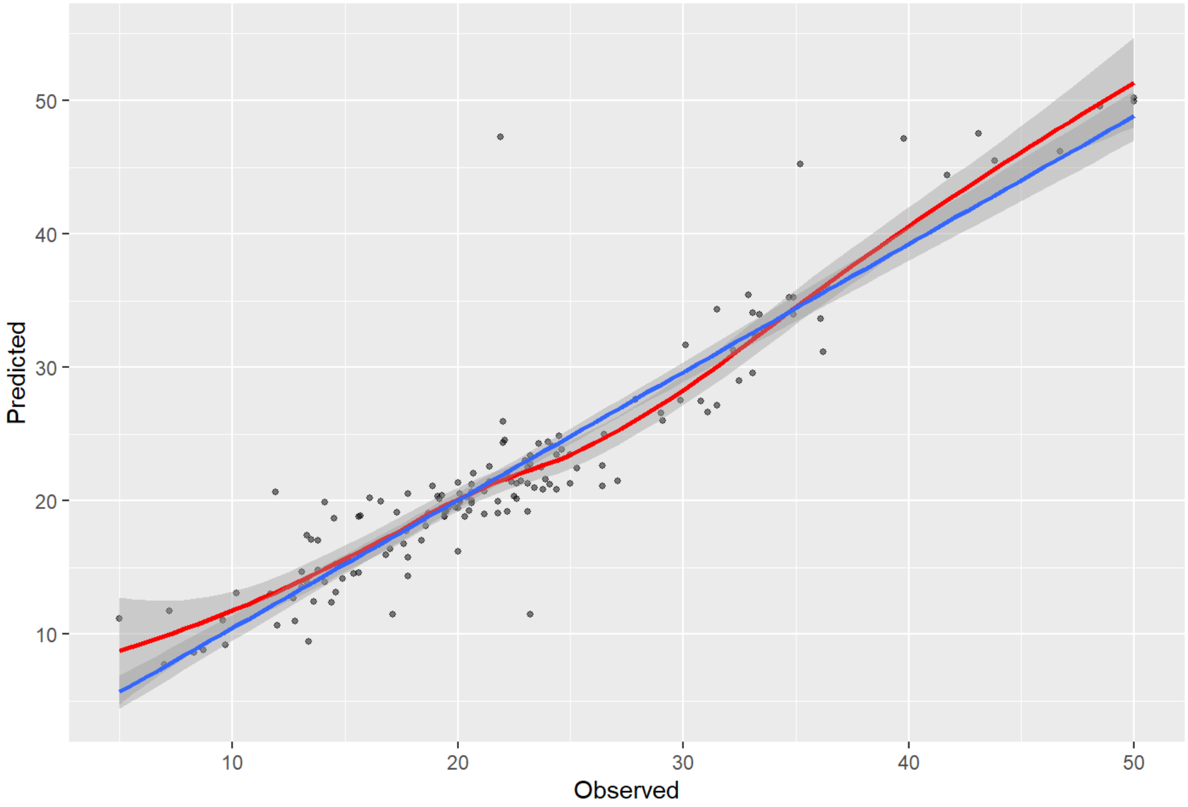
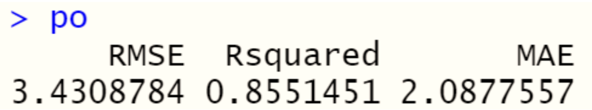
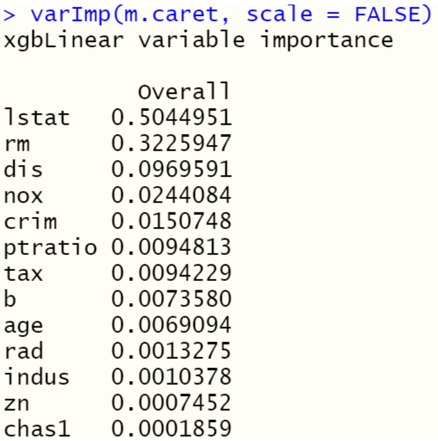


Figure 9: Predicted v/s observed values of house price prediction model before tuning



```
> po
     RMSE  Rsquared       MAE
3.4308784 0.8551451 2.0877557
```

Figure 10: Measures of fit of initial house price prediction model



```
> varImp(m.caret, scale = FALSE)
xgbLinear variable importance

          Overall
lstat   0.5044951
rm      0.3225947
dis     0.0969591
nox     0.0244084
crim    0.0150748
ptratio 0.0094813
tax     0.0094229
b       0.0073580
age     0.0069094
rad     0.0013275
indus   0.0010378
zn      0.0007452
chas1   0.0001859
```

Figure 11: Importance of all features in the model

The variable importance of the final model was checked and shown in Figure 11. To try feature selection, the least important variables ‘chas’ and ‘zn’ were removed from the model since they had values less than 0.001. Moreover, ‘zn’ had numerous outliers, and ‘chas’ was a categorical variable. However, even after removing those variables, the model trained with the 11 remaining variables, provided the same measures of fit. Hence all 13 predictor variables were included to create the final Boston house price prediction model. Since the final model gave excellent measures of fit, it performs well in house price prediction.

However, there are many limitations to this model. It assumes a linear relationship between predictor variables and target variable, which may not be always true. Several attributes having outliers and skewed distributions were used for model training, which would affect the model performance. The data used in this model was collected in the 1970s, and as a result, the values of crime rates, taxes, population status, etc., would have changed with time. The dataset is too small, which increases the likelihood that it will be biased.

This model can be improved in the future by including more relevant attributes for housing in the current scenario, such as proximity to public transport, the terrain of the property, the energy efficiency of the building, etc. This model can be compared with some other models, such as GBM, XGBTree, etc. The outliers can be capped and the model can be trained to get better performance.

REFERENCES

Bertolini, R., 2021. *Evaluating performance variability of data pipelines for binary classification with applications to predictive learning analytics* (Doctoral dissertation, State University of New York at Stony Brook).

Breiman, L., 1996. *Bias, variance, and arcing classifiers*. Tech. Rep. 460, Statistics Department, University of California, Berkeley, CA, USA.

Chen, T. and Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Comber, L. and Brunsdon, C., 2020. *Geographical data science and spatial data analysis: an introduction in R*. Sage.

Dalal, S., Seth, B., Radulescu, M., Secara, C. and Tolea, C., 2022. Predicting Fraud in Financial Payment Services through Optimized Hyper-Parameter-Tuned XGBoost Model. *Mathematics*, *10*(24), p.4679.

FRĠEDMAN, J., 2002. Stochastic gradient boosting. Computational statistics and data analysis.

Harrison Jr, D. and Rubinfeld, D.L., 1978. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, *5*(1), pp.81-102.

Jiang, X., Jia, Z., Li, L. and Zhao, T., 2022. Understanding Housing Prices Using Geographic Big Data: A Case Study in Shenzhen. *Sustainability*, *14*(9), p.5307.

Leisch, F., Dimitriadou, E., Leisch, M.F. and No, Z., 2009. Package ‘mlbench’. *CRAN*.

Pesantez-Narvaez, J., Guillen, M. and Alcañiz, M., 2019. Predicting motor insurance claims using telematics data—XGBoost versus logistic regression. *Risks*, *7*(2), p.70.

Xenochristou, M. and Kapelan, Z., 2020. An ensemble stacked model with bias correction for improved water demand forecasting. *Urban Water Journal*, *17*(3), pp.212-223.