Statistical learning: Moddling the common household price

Songze Yang, u7192786

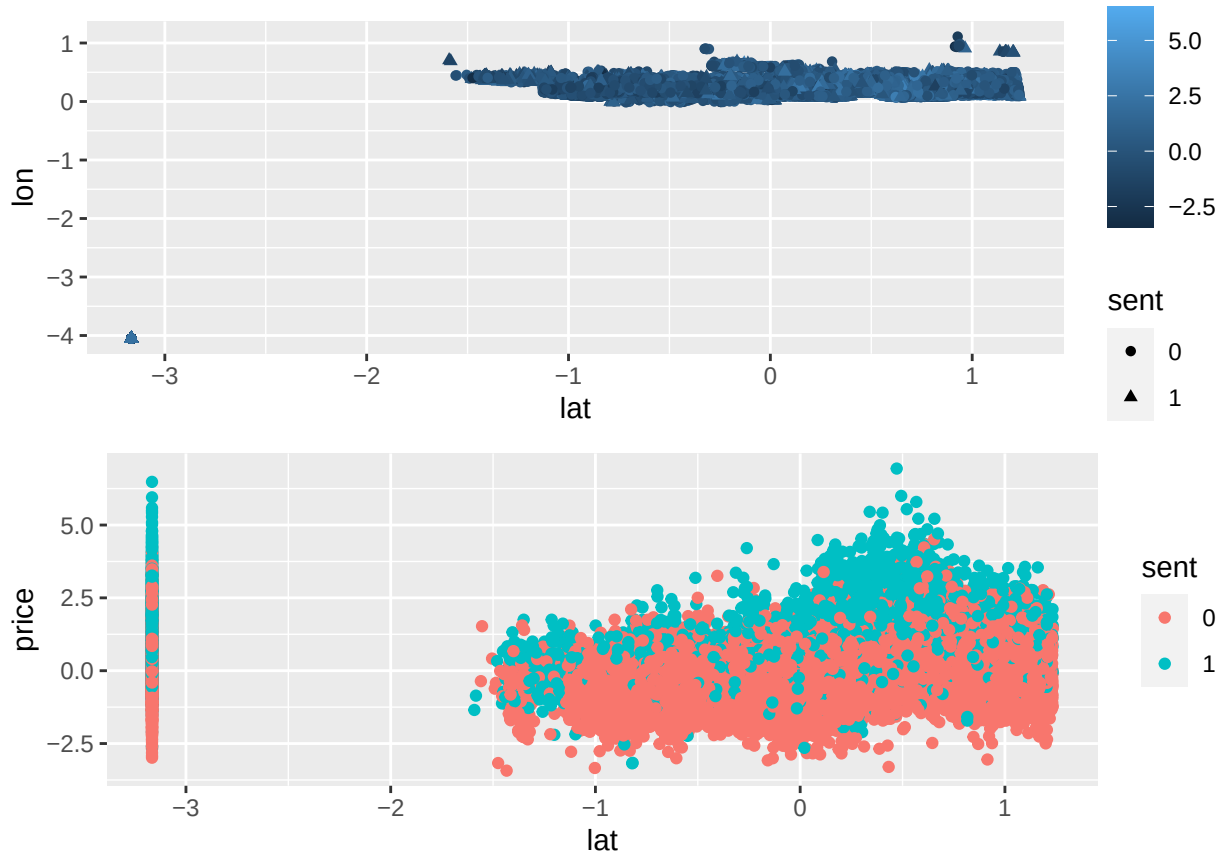
Introduction

The household information of the Australian city of Ken-Behrenia is provided in the training and test dataset, which have 30,000 and 10,000 observation.

There are 12 covariates in the dataset, including prior sale condition measurement score [cond], the sold year [year], the decade the house built [built], the location of the house in latitude [lat] and in longitude [lon], square meters of living space [sq.m.h] and of the lot size [sq.m.block] and of the pool size [sq.m.pool], whether renovated prior to sale [reno], number of bedrooms [bedrooms] and of bathrooms [bathrooms] and density measurement score [environ].

Two responses are in the data, namely the the scaled sale price [price] and the satisfaction measurement [sent].

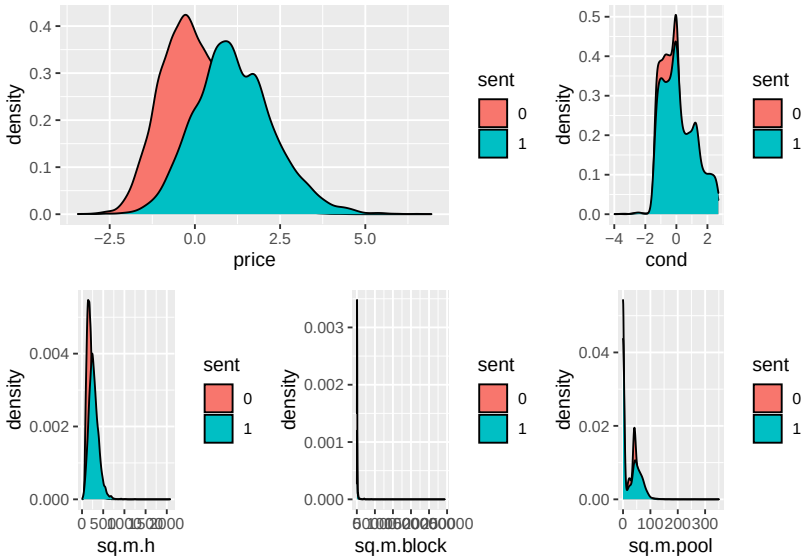
Exploratory data analysis



The map of the household distribution is shown above with color to denote different price of household and circle to denote high sentiment and triangle to denote medium and low sentiment. A number of 21915 the households are grouped from upper right corner. Additionally, a number of 2148 suburb is located in the left bottom corner about -4 in longitude and -3 in latitude. A reasonable explanation is that the larger area may be the main residual area of the Ken-Behrenia, which includes 91.07% of the data, while the smaller suburb covers 8.92%, both in case of ignoring the missing value. The test data has similar feature, where two distinct groups emerge. Also, the price and sent are randomly distributed in different locations from the graph.

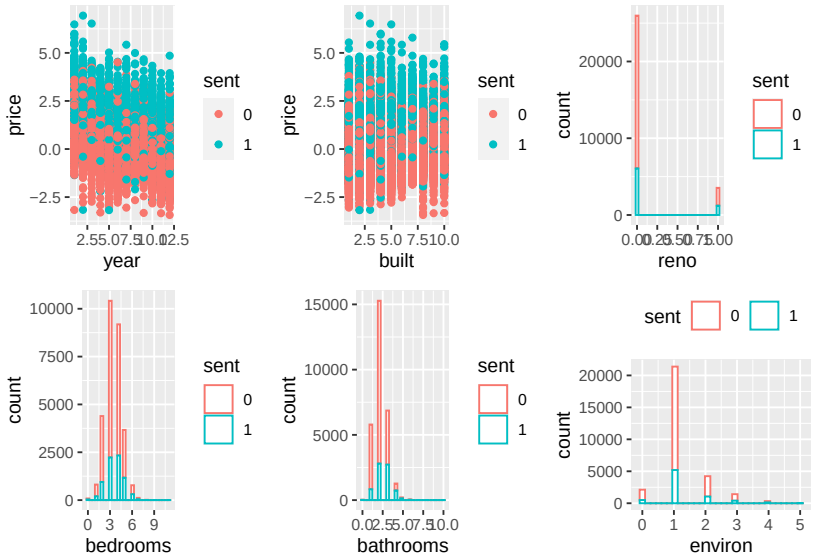
It is common belief that the latitude of the house is associated with the household price. Reasons behind this may be better sunlight, warmer climate and enjoyable indoor swimming pool. However, our data shows that there is no strong relationship between the the latitude and price. But higher price may be related to higher sentiment score.

Excluding the ID for each house, the distributions of five continuous variables with sent are shown below, namely the price, condition score, and the square meter of living space, lot size and pool.

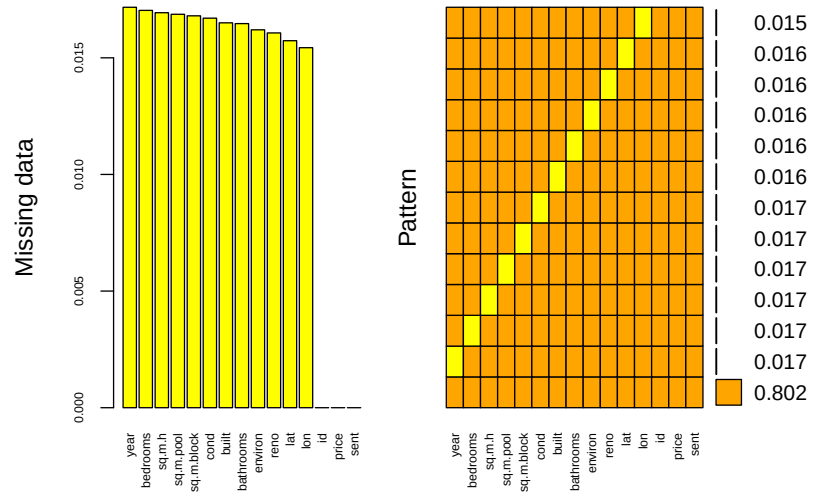


The price and sent are positively associated, confirmed our previous conclusion. As for condition, the houses with -1 to 0 score have more lower sent (0). Concerning the area of living space, lot size and pool size, the significant skewness discerned through the square meter variables can result in the subsequent model's classification being biased. Also, the predictor variable price in our case is standardized. Thus, a standardization of our square meter variable applies here.

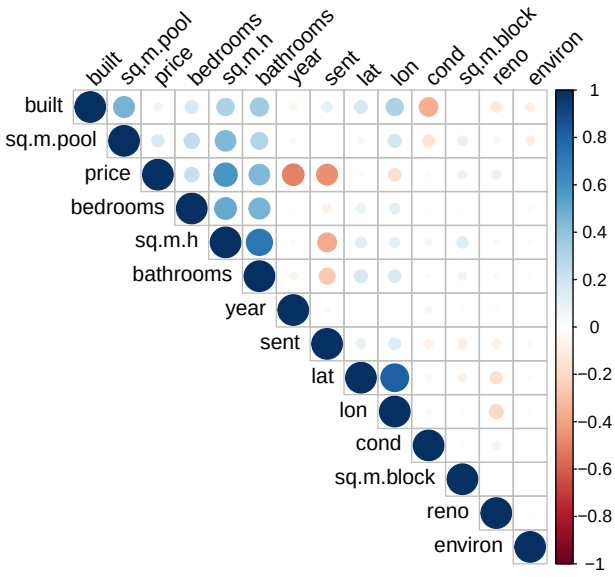
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



With regards to discrete data, namely reno, bedrooms, bathrooms and environ are right skewed. The distribution of sent is fairly constant though years and decade household built in. The skewness is fine in the linear model but may cause problem in the deep learning part. We apply scale function in that part.



There are approximately 19.8% of our data point contains missing values. Apart from id and our predictors, covariates have nearly 0.016% missing value below 2%. Thus, a multiple imputation apply here.

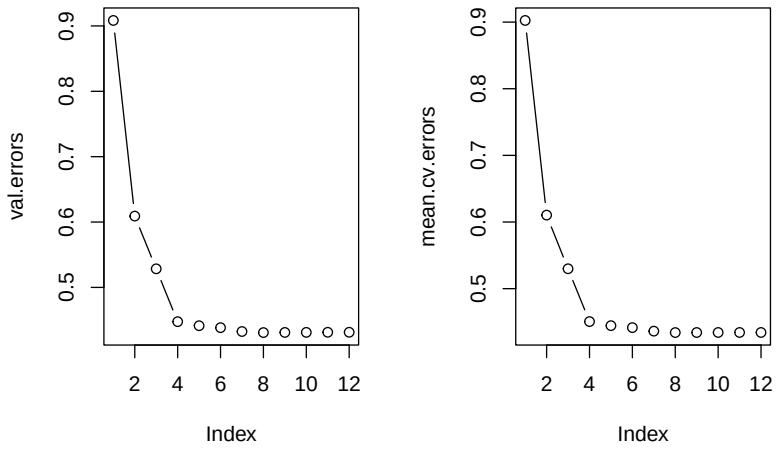


The correlation matrix is presented above. Only the lat and lon are considered as highly correlated with correlation higher than 80%. High correlation causes a compromised imputation result. Thus, we will consider a predictive mean matching approach to reduce this correlation.

Modelling the household price

linear regression

The most common way to tackle a regression class question is linear regression modelling. We will rely on the regsubsets function to select each best model processing a certain number of covariates. The regsubsets function will select the model with lowest R squared in each step. Then we will use validation set and 10-fold cross validation to test the prediction power of our model. The plotted results is plotted below.



Both methods suggest the number 8 model. So our model can be written as follows. The validation MSE for this model is 0.4309205 and 10-fold validation MSE is 0.4341238.

$$price = 1.36534177 - 0.14954392 * year - 0.02272077 * built + 0.44156831 * lat - 0.57386103 * lon +$$
$$0.62459880 * sq.m.h + 0.05210058 * sq.m.block - 0.09477073 * bedrooms + 0.13139679 * bathrooms$$

Ridge and Lasso Regression

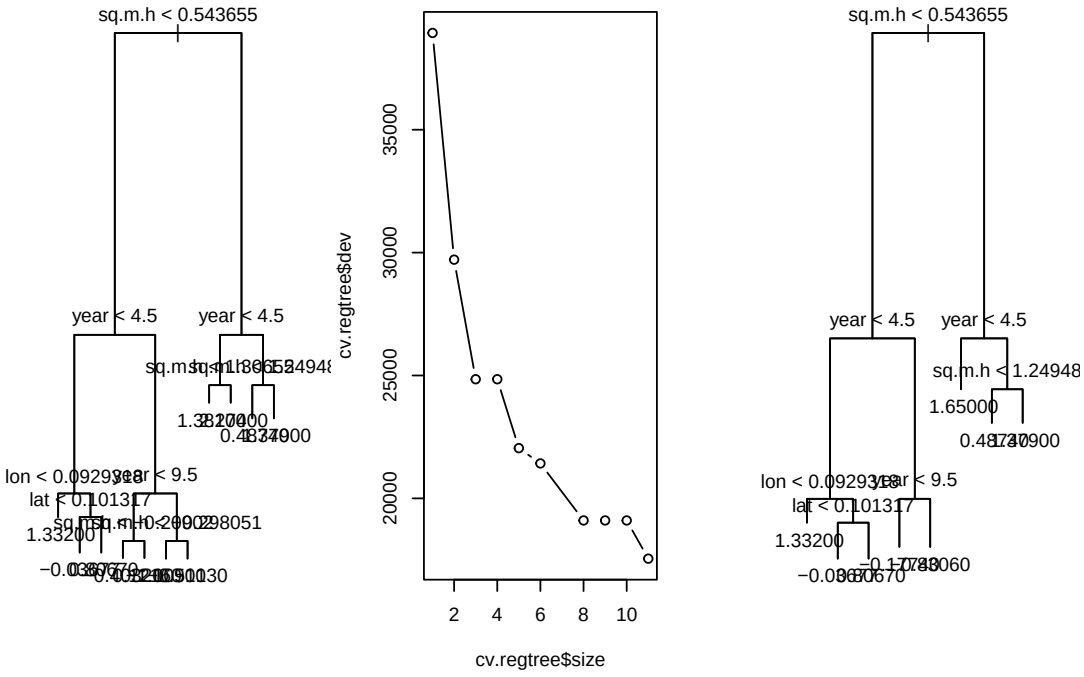
The shrinkage method is a family of important method in improving performance from simple linear regression. We will select dimension reduction methods from 2 class of model, namely ridge regression, lasso regression. In both ridge and lasso regression, the turning parameter is selected by the k-fold validation. Thus, the 10-fold is applied here.

The 10-fold validation result is 0.443308 for ridge regression and 0.4342615 for lasso regression. The lasso regression improves slightly than the ridge regression.

Decision Tree

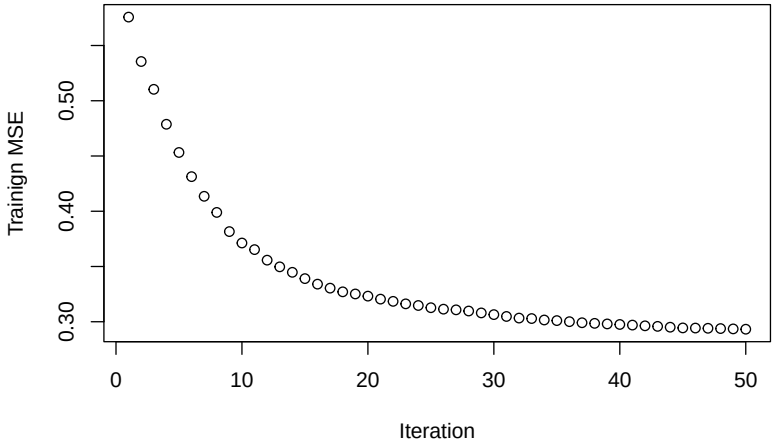
In the previous analysis, the interaction between covariates is not measured. The linear regression model assumes that the predictors in the model are independently distributed, a assumption hardly achieved in the real setting. However, the tree model handles this problem smoothly. Next, we will apply a number of tree models.

A normal regression tree stratifying or segmenting the predictor space into a number of simple regions. A set of rules are used in splitting the predictor space. To determine variables linked to response, a regression is fitted and prune to prevent overfitting. Applying a 10-fold cross validation, the MSE is not reduced after the tree has 8 nodes. We choose the tuning parameters alpha based on the CV result. The pruned part mainly come from the sq.m.h variable split in each subtree after a initial splitting of sq.m.h. The 10-fold cross-validation score is 19051.64. The performance of decision-tree is very low compared to linear and shrinkage method.



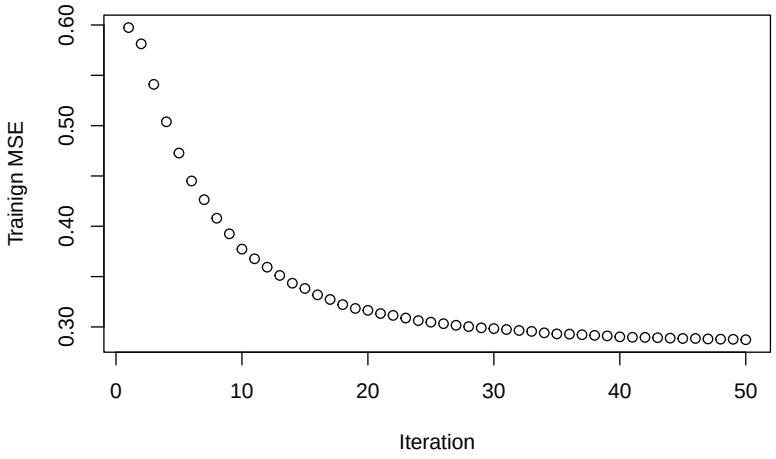
Bagging

To improve this, bagging algorithm is applied here. The general-purpose bagging algorithm is used to reduce the variance. It involves generating B number of bootstrap sample, fitting trees on each bootstrap example and averaging the results of many trees. As the bootstrap have cooperated into the bagging algorithm, we do not need to perform the cross validation. However, to avoid overfitting, We can choose the number of tree to grow from the graph below. The 100 trees model follows no significant improvement in the accuracy. The training MSE at this point is 0.2850119.



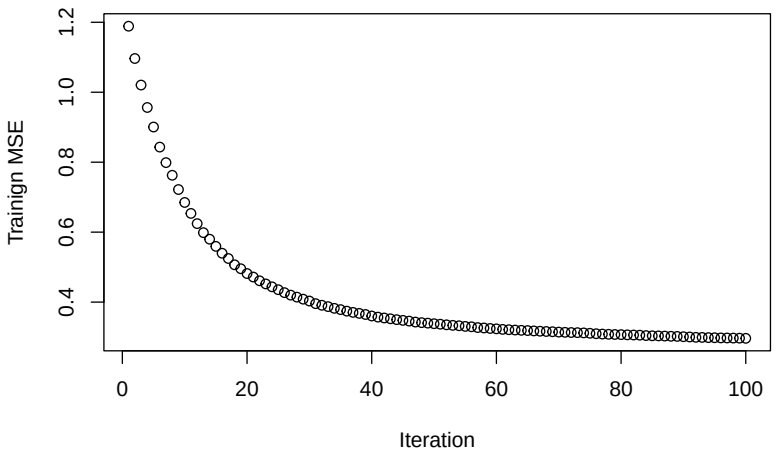
Random Forest

Next, random forest combines the bagging method to further improve the prediction power. It manages to reduce the MSE by selecting a subset of m covariates when fitting each node. Thus, the correlation between trees is tweaked. The m here is choosed by the square root of p, where p is the number of covariates in the data. We then need to choose the number of trees to prevent overfitting. The 50 tree model has related low MSE as in the graph. The training MSE at this point is 0.2764581.



Boosting Tree

Lastly, the sequential boosting algorithm can further push our prediction power. Boosting method update the tree based on the previous tree. It repeatedly targets on the residual from the previous tree and add new tree to improve the fitting. It prevents overfitting naturally by the shrinkage parameter, which is by default 0.1. We will fit a 100 trees model with interaction.depth = 4. As overfitting will be mild in boosting, we will iterate through to get a better prediction accuracy.



The MSE is stable after 50 iteration. The MSE at this point is 0.3382037. The number of tree is the same as that in random forest.

Xgboost

The boost tree method nowadays has been pushed to its edge. The gradient boosting framework has been cooperated into the boost decision tree model to reduce error and to work faster. Many open sourced software libaray improve our performance in scoring, such as XGboost, lightGBM and Catboost. The three are very similar but with some difference in tree-growing strategy. We will use Xgboost to do a horse race with the algorithm above.

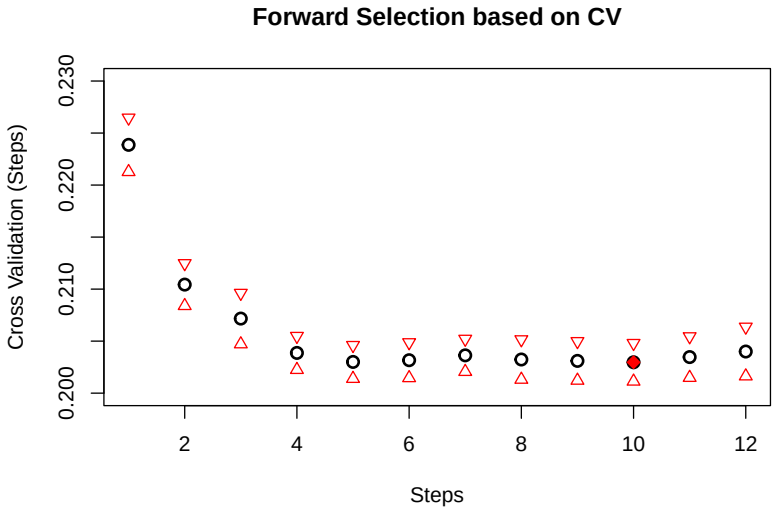
We will only use the default parameter to train the model and use a 10-fold cross validation to select the best iteration. There are many parameter we can tune in train a Xgboost model, mainly booster parameters, learning task parameters and general parameters. Due to the certain time limit in the kaggle competition, we will not use algorithm, such as grid search, to tune our parameter to get a better prediction. However, ways do exist to improve our model. For example, the decrease of learning rate eta from the default value 0.3 to 0.03 can potentially give us a better model. Or the tuning of colsample_bytree parameter can generate more randomness at each node thus to give us a random forest like boosting tree. Some package can help us tune the parameter here, for instance, the MLR package.

The model 10 fold training MSE is 0.2474197, which is calculated by the the RMSE provided by the output. The test MSE on kaggle is 0.25146, which is currently our best model for price. The algorithm is much efficient than the boost tree before.

Modelling the satisfaction score (sent)

logistic regression

The logistic regression is by far the most common technique to tackle the binary classification problem. It assumes a log of odd link function to predict the probability of each class. Prior to fit a logistic model to data, a forward selection based on training MSE is used to shrink our predictor space. Also, while we apply 10-fold cross validation to compute MSE for our model, the confidence interval is computed and graphed as well. The result is shown below.

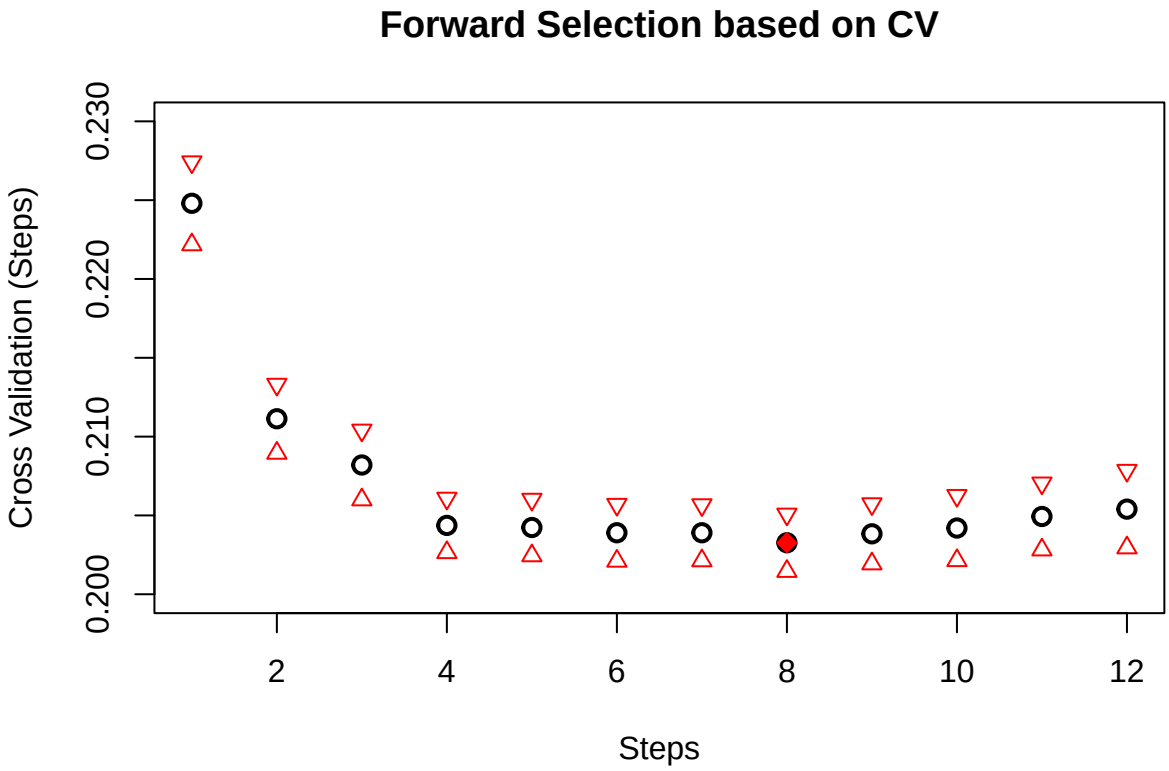


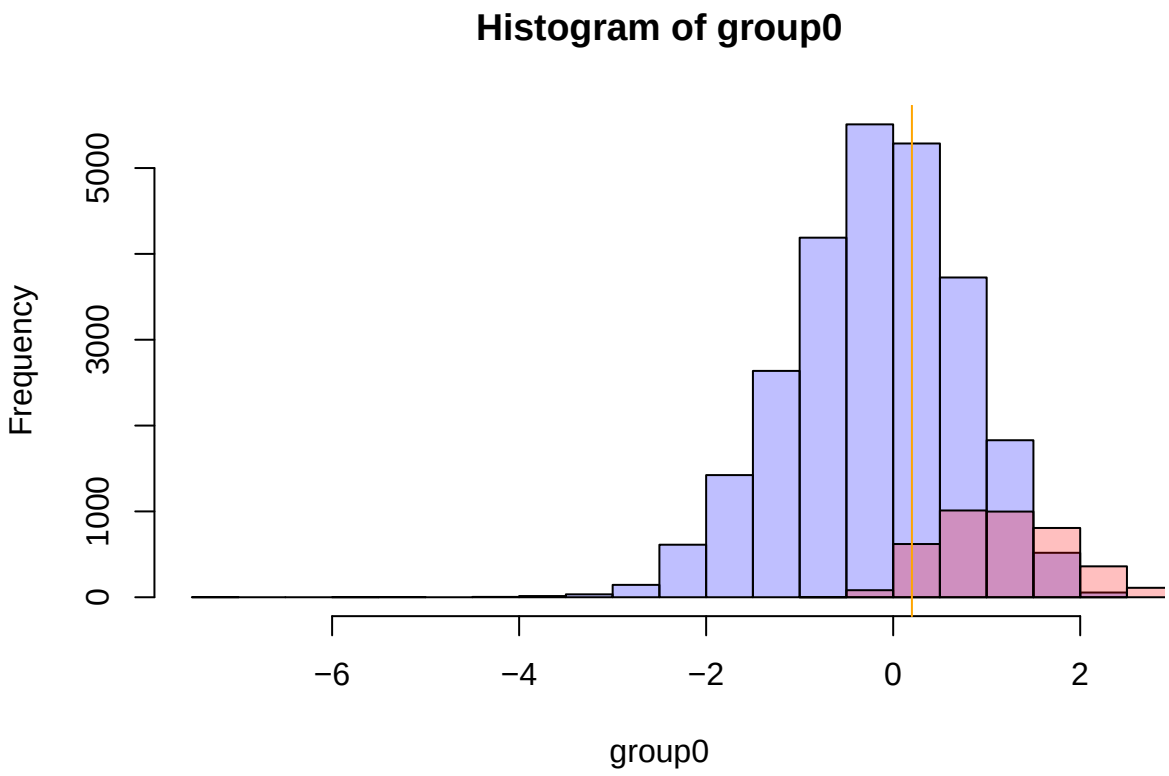
The result shows that after 4 initial variables are added into the model, the training MSE do not reduce significantly. Thus, our final model is fitted as below. The training MSE for this model is 0.2038667.

$$log(\frac{p(sent=1)}{1-p(sent=1)}) = -0.9077081 - 1.5070861 * sq.m.h + 0.2241024 * built+$$
$$0.2924656 * lon + 0.3444135 * bedrooms$$

linear discriminant analysis

LDA assumes that all independent variables are drawn from a multivariate Gaussian distribution. Our data have left or right skewed problems seen in the exploratory data analysis. Subsequent to a standardization of our data, our data is ready to fit a LDA model. As before, we will apply the same procedure using the forward selection algorithm to perform variable shrinkage.


Forward Selection based on CV
Cross Validation (Steps)
0.200
0.210
0.220
0.230
2
4
6
8
10
12
Steps

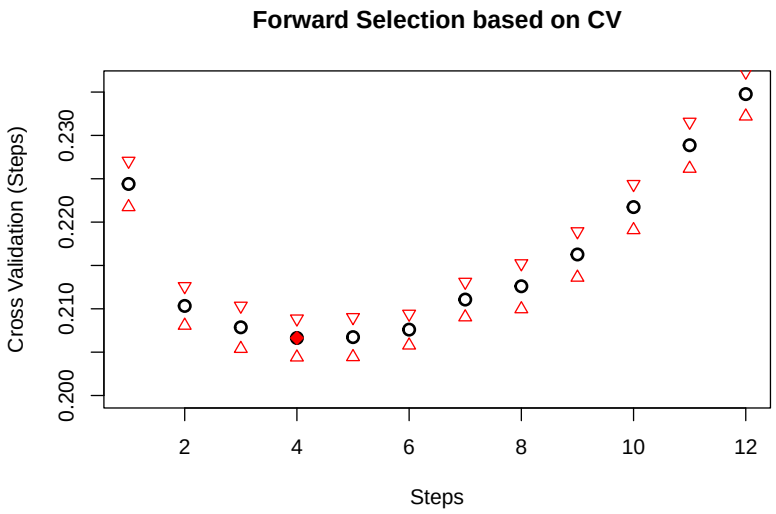


The result is similar to the logistic regression. After the initial four variables are added into the model, the MSE of data stay in a flat range. Thus, the model with four variables, namely sq.m.h, built, bedrooms and lon, is fitted. The training MSE for this model is 0.2038667.

Quadratic discriminant analysis

Alike LDA, QDA assumes that predictor variables are normally distributed and have minimal multicollinearity. Furthermore, the QDA assumes that each variable possess their own variance-covariance matrix. In our case, the variables are either left or right skewed but the variance for each predictor are unique. Thus, QDA is appropriate.

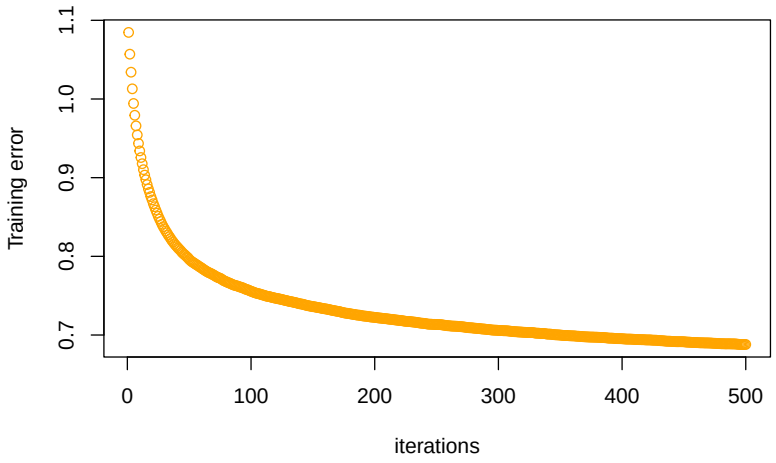
```
## [1] "sent ~  + sq.m.h + built + bedrooms + sq.m.pool + year + cond + bathrooms + environ + reno + la
```



The result is close to the linear discriminant analysis. After the first four variables are added into the model, the MSE of data climbs up, suggesting a overfitting. Thus, the model with four variables, namely sq.m.h, built, bedrooms and sq.m.pool, is fitted. The training MSE for this model is 0.2066333.

Boost Tree Model

From the previous analysis, the boost tree method combines the advantage of bagging and random forest. To make the best prediction, the boost tree is used in the classification of sent. The prior distribution is bernoulli for two class classification and the number of tree to try is set at 500. The interaction depth is fitted at 4.



After 500 iteration, the training error reduces very slowly. Thus, we will use 500 as number of iteration to prevent overfitting. The MSE at this iteration is 0.6879958.

Xgboost model

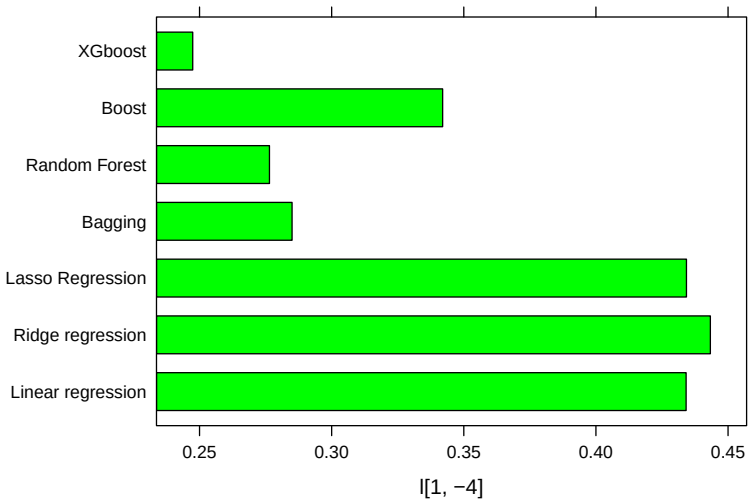
The open-source software XGboost can also be used. As before, the MSE is the lowest amount all the model.

Conclusion

Regarding the price, the year, built, lat and lon are 4 most important factor, as it leads to the most explaination in the forward selection process for linear model. The lasso regression suggests that the cond, year, built and lat are most import variable. The lasso do not shrink any of the variables to 0, suggesting that the other variables may take effect in predicting. The tree suggests that the sq.m.h, year, lon and lat is important. There is some discrepancy between but all points to the year the household sold(timing of the house sold), the location (lat and lon variable) and area of the house(sq.m.h). This variables match our real life experience.

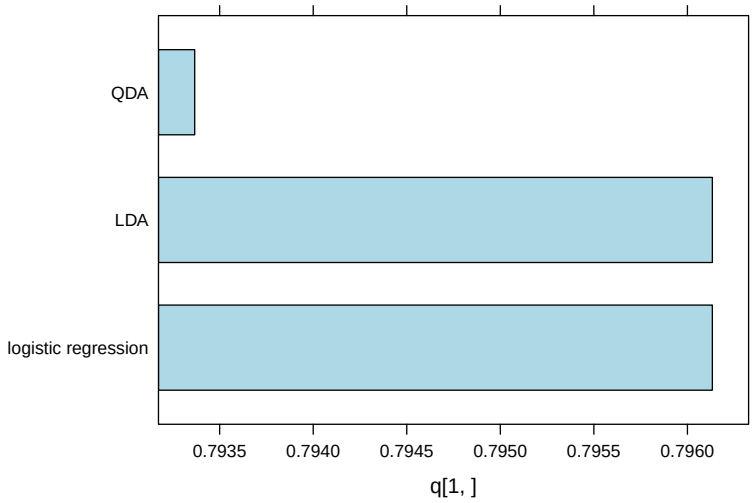
In terms of the predictivity, the linear family, that is linear regression, ridge regression and lasso regression, produces similar prediction accuracy, all centered around 0.43. For the tree family, the improvement makes from bagging to random forest. The boosting model with 100 iteration underperforms the random forest and bagging models. Like above analysis, the prediction power of boosting will eventually overtake other tree models, as suggested by the improvement led by XGboost model. All the MSE mentioned here is measured by the 10-fold cross validation on the training data.

```
## Warning in barchart.numeric(l[1, -4], colnames(l[1, -4]), col = "green"):
## explicit 'data' specification ignored
```



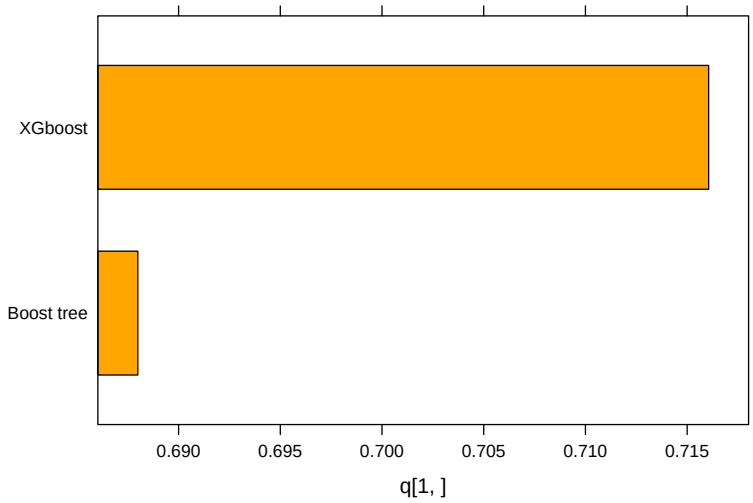
For the prediction of sent, similarly the logistic regression and LDA return approximate correct rate. The QDA assumes that variables have non-identical variance, assumption that accords to our data. But the correct rate for this model is lower than both logistic regression and LDA. This may be caused by the nature that our data is linearly separable.

```
## Warning in barchart.numeric(q[1, ], colnames(q[1, ]), col = "lightblue"):
## explicit 'data' specification ignored
```



Likewise, the Xgboost provides better fit than normal boost model. The better parameter and computational algorithm helps the model to achieve better accuracy in shorter time.

```
## Warning in barchart.numeric(q[1, ], colnames(q[1, ]), col = "orange"): explicit
## 'data' specification ignored
```



Lastly, the sent variable is associated with sq.m.h the most, since the logistic regression, LDA and QDA all select it in the first place. Other most important factors are built, bedrooms and sq.m.pool. For the three model here, we assumes that the covariates are independent, meaning that for certain house size, bigger pool and more bedrooms may be more attractive on the market. This is my scientific answer to the problem.