Midterm Report

Chirag Shah, Nathalie Fadel, Deepika Dilip

Introduction

While median wages are remaining stagnant, housing prices in Taipei, Taiwan are increasing exponentially. Using a historical data set of real estate valuation from the Sindian District and New Taipei City in Taiwan, we decided to explore which predictors had the most impact on housing prices in Taipei.

We outlined our analysis plan so that our models excluded observations that had missing values. However, we did not have any observations with missing information and therefore a sensitivity analysis was not necessary. The variable "no" in the original dataset was dropped, as it was an observation id, which would not be relevant in our analysis.

This dataset contains 414 observations, with the 6 predictors as follows:

- X1=the transaction date (for example, 2013.250=2013 March, 2013.500=2013 June, etc.)
- X2=the house age (unit: year)
- X3=the distance to the nearest MRT station (unit: meter)
- X4=the number of convenience stores in the living circle on foot (integer)
- X5=the geographic coordinate, latitude. (unit: degree)
- X6=the geographic coordinate, longitude. (unit: degree)

The outcome is as follows:

- Y= house price of unit area (10000 New Taiwan Dollar/Ping, where Ping is a local unit, 1 Ping = 3.3 meter squared)

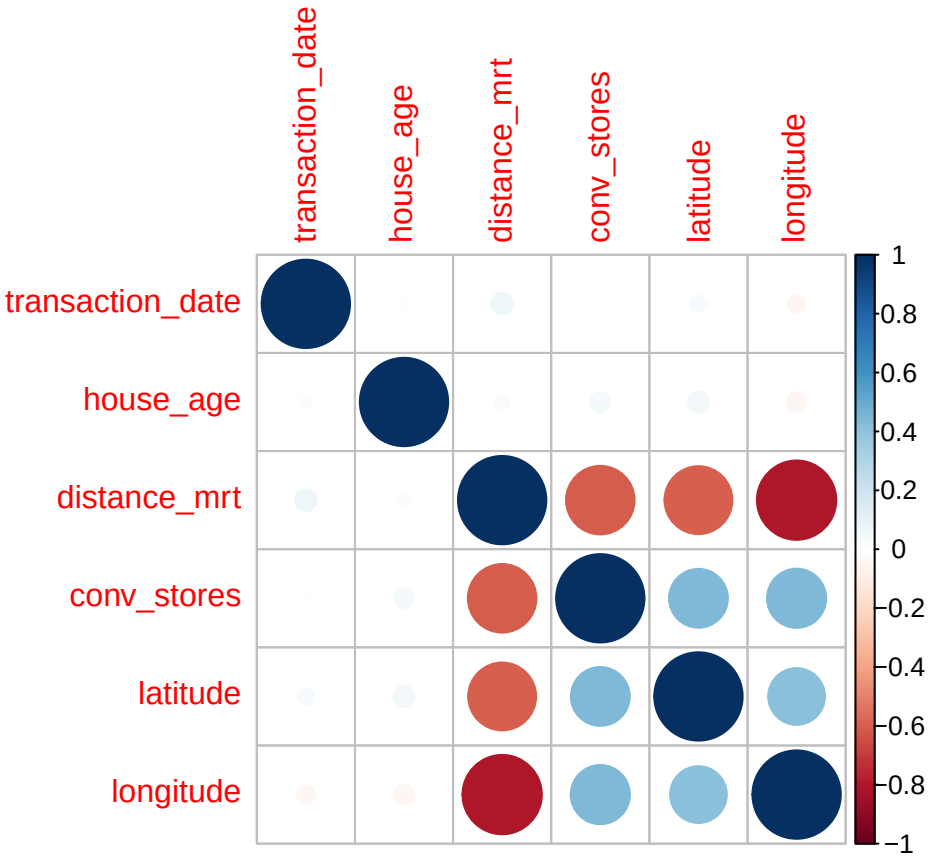
Exploratory Data Analysis

Descriptive Statistics

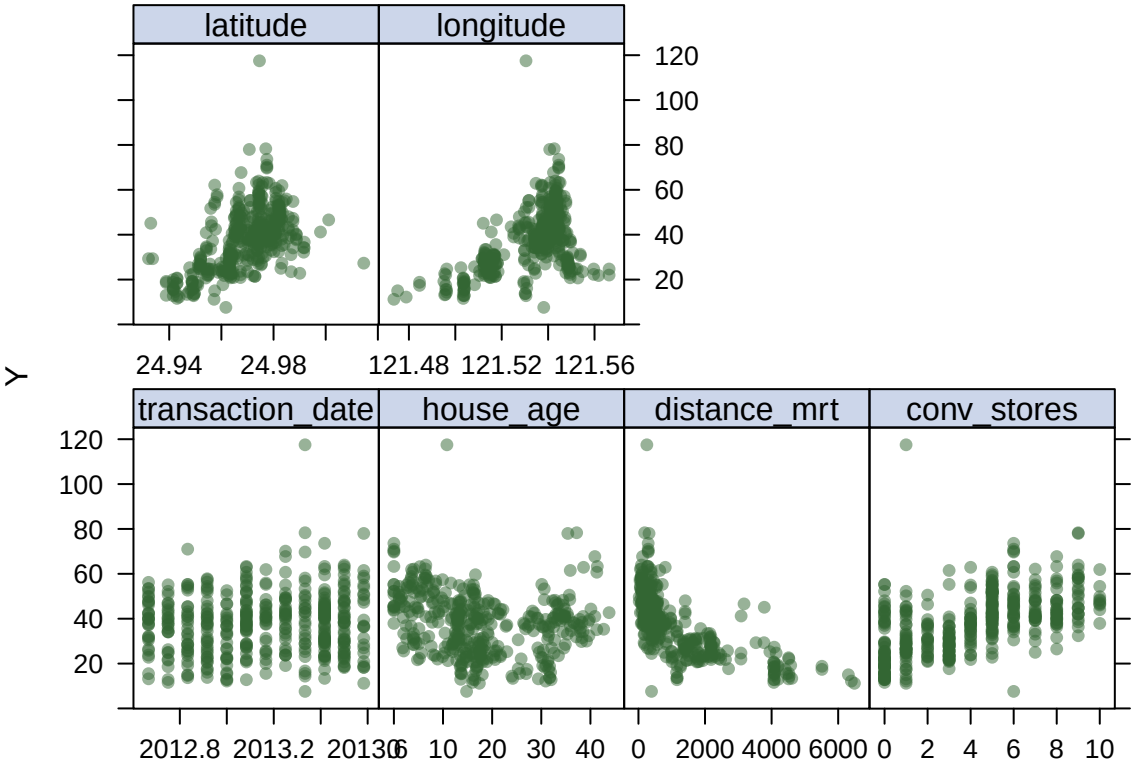
	transaction_date	house_age	distance_mrt	conv_stores	latitude	longitude	house_price
nbr.val	414	414	414	414	414	414	414
nbr.null	0	17	0	67	0	0	0
nbr.na	0	0	0	0	0	0	0
min	2013	0	23	0	25	121	8
max	2014	44	6488	10	25	122	118
range	1	44	6465	10	0	0	110
sum	833444	7333	448729	1695	10337	50315	15724
median	2013	16	492	4	25	122	38
mean	2013	18	1084	4	25	122	38
SE.mean	0	1	62	0	0	0	1
CI.mean.0.95	0	1	122	0	0	0	1
var	0	130	1592921	9	0	0	185
std.dev	0	11	1262	3	0	0	14
coef.var	0	1	1	1	0	0	0

Designating the Predictors and Outcome

Observing Correlation of Predictors



Scatterplot



We can see that there may be a non-linear relationship between house price and distance to MRT station, and between house price and house age. We will test both of these terms independently and together as spline terms in GAM models. Transaction date and distance to convenience stores appear to be categorical predictors.

Models

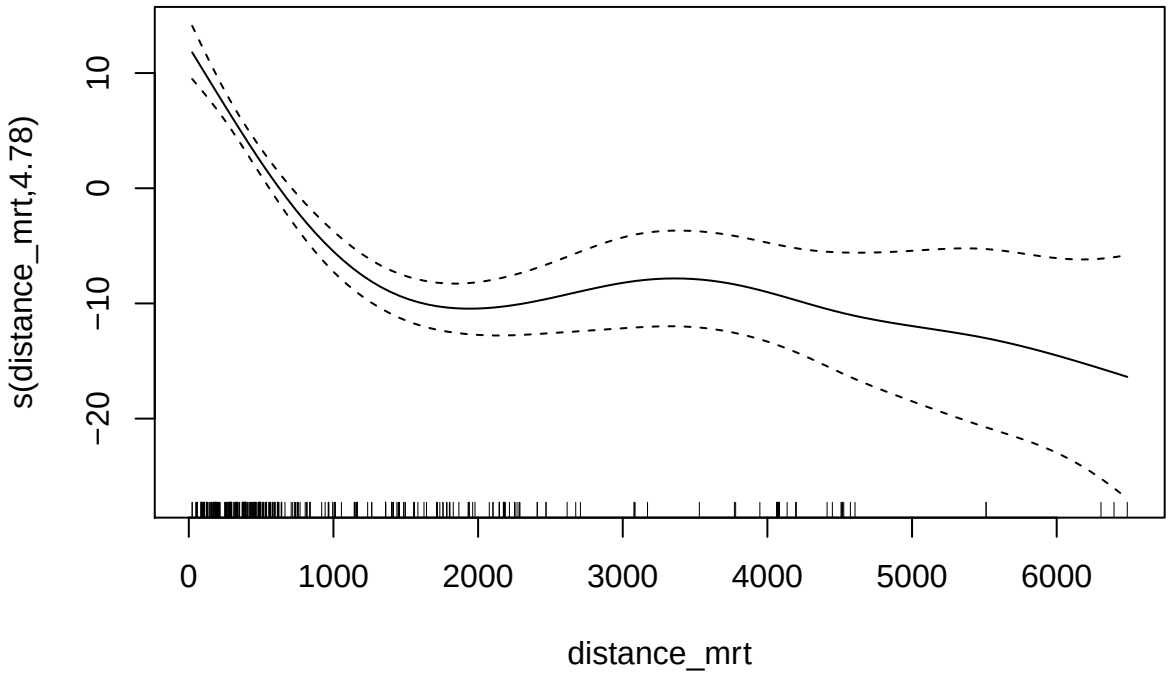
We used a total of six different modeling techniques to estimate house prices. GAMS, MARS, and KNN were nonparametric while ridge regression, lasso regression, and linear regression were parametric. We started with the most flexible model and progressed to the most stringent. All six predictors mentioned in the Introduction were used in our analysis. The results are summarized in the following table:

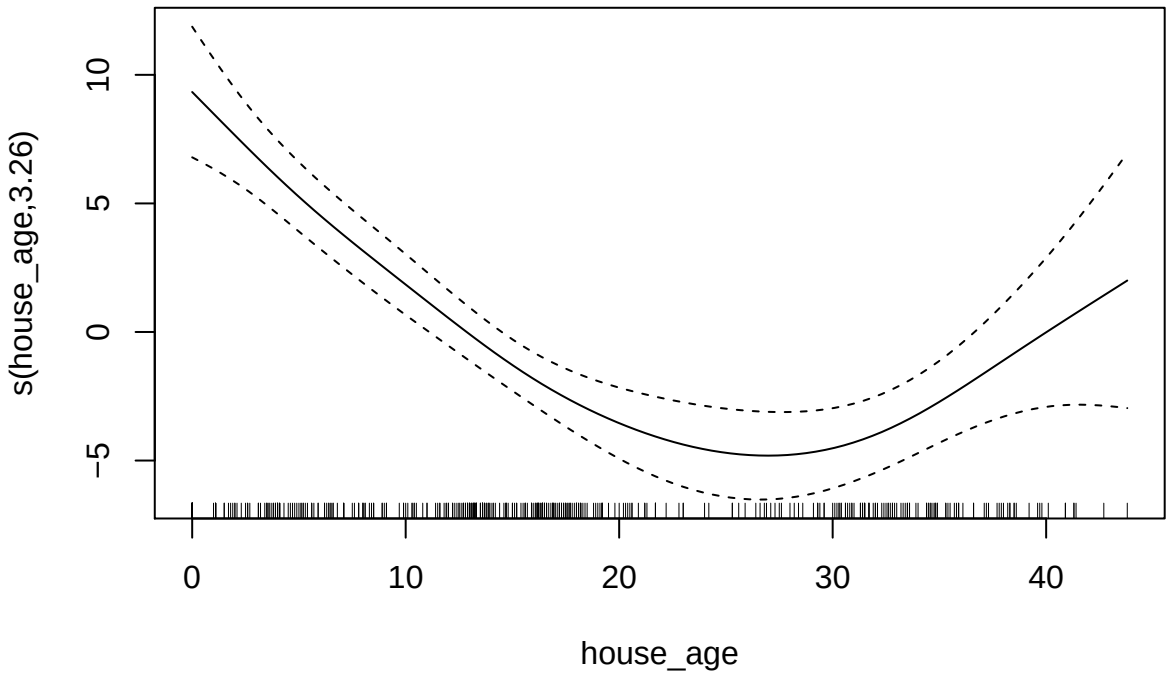
Summary of Models Used

Model	R^2	RMSE	Model Limitations
GAM	0.687	Content Cell	Content Cell
MARS	0.7005081	7.568432	Content Cell
KNN	xxxxxx	8.208905	Content Cell
Ridge Regression	xxxxxx	8.799873	Content Cell
Lasso	Content Cell	8.810508	Content Cell
Linear Regression	0.5824	8.782313	LINE assumption needs to hold

GAM

```
## Analysis of Deviance Table
##
## Model 1: house_price ~ transaction_date + house_age + distance_mrt + conv_stores +
##     latitude + longitude
## Model 2: house_price ~ transaction_date + house_age + s(distance_mrt) +
##     conv_stores + latitude + longitude
## Model 3: house_price ~ transaction_date + s(house_age) + distance_mrt +
##     conv_stores + latitude + longitude
## Model 4: house_price ~ transaction_date + s(house_age) + s(distance_mrt) +
##     conv_stores + latitude + longitude
##   Resid. Df Resid. Dev      Df Deviance      F    Pr(>F)
## 1    407.00      31931
## 2    402.17      24791  4.8295   7140.1 25.515 < 2.2e-16 ***
## 3    403.93      29039 -1.7641  -4247.5 41.552 1.487e-15 ***
## 4    399.93      23273  4.0061   5766.0 24.839 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

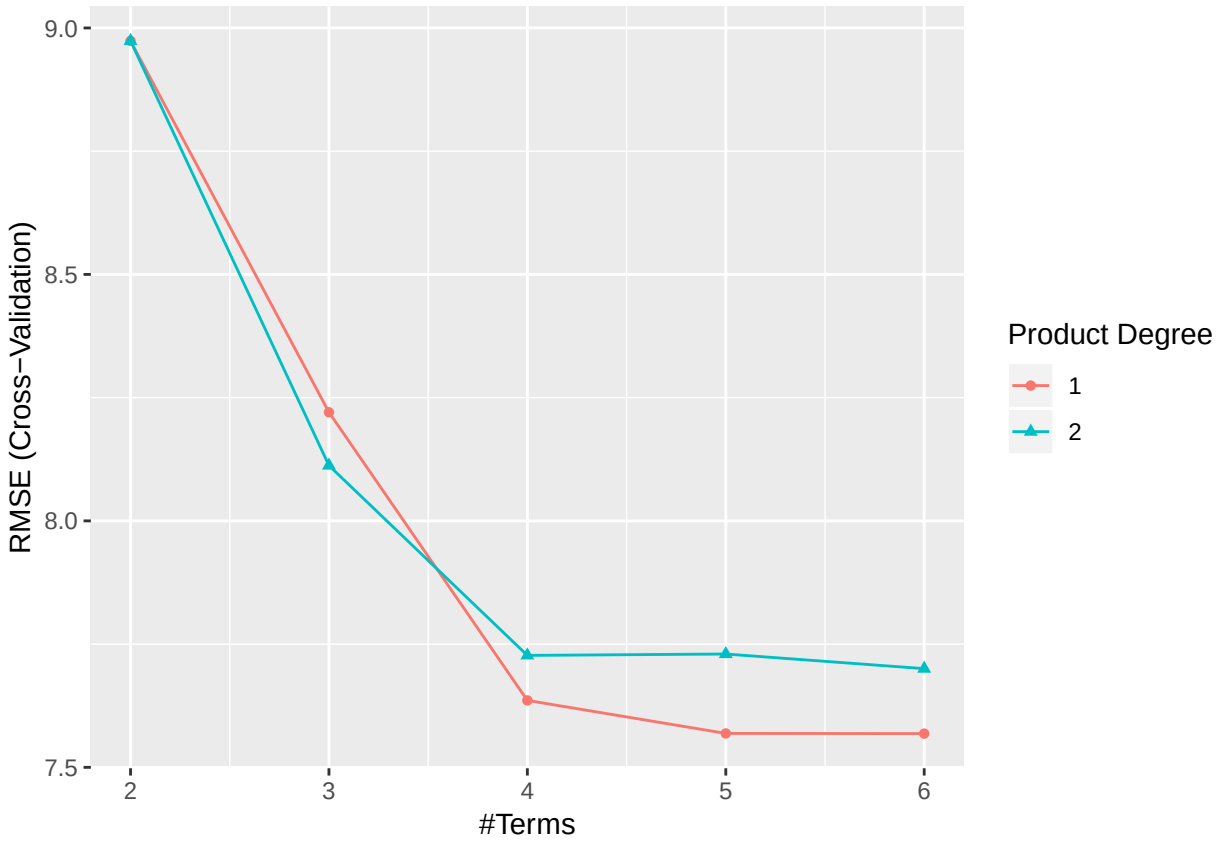




GAM M4 has best R^2 value. The model uses house age and distance to the nearest MRT station as spline predictors, while also including transaction date, latitude, longitude, and the number of convenience stores in the living circle on foot.

MARS

Assumptions:




```
##   nprune degree
## 5      6      1
```

```
##                 (Intercept)     h(1144.44-distance_mrt)
##                18.63760631                  0.01878469
##           h(27.1-house_age) h(2013.42-transaction_date)
##                 0.36442551                 -6.50921515
##        h(121.545-longitude)         h(latitude-24.9415)
##                -73.67130513                304.15900213
```

```
## Multivariate Adaptive Regression Spline
##
## 414 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 372, 371, 372, 374, 372, 374, ...
## Resampling results across tuning parameters:
##
##   degree  nprune  RMSE      Rsquared   MAE
##   1       2       8.973708  0.5724324  6.564668
##   1       3       8.220322  0.6415487  5.847956
##   1       4       7.635796  0.6938029  5.081914
##   1       5       7.568810  0.7008981  5.102291
##   1       6       7.568432  0.7005081  5.162027
##   2       2       8.973708  0.5724324  6.564668
##   2       3       8.112254  0.6544432  5.736867
```

```
##   2       4       7.727124  0.6879763  5.249256
##   2       5       7.729851  0.6897473  5.258304
##   2       6       7.700241  0.6921679  5.233592
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nprune = 6 and degree = 1.
```

In order to minimize the MSE, 1 degree of interaction and 6 retained terms were used (as depicted on the plot). The most important predictors are depicted below:

KNN

Assumptions:

```
trainX <- train[,names(train) != "house_price"]
preProcValues <- preProcess(x = trainX,method = c("center", "scale"))
preProcValues
```

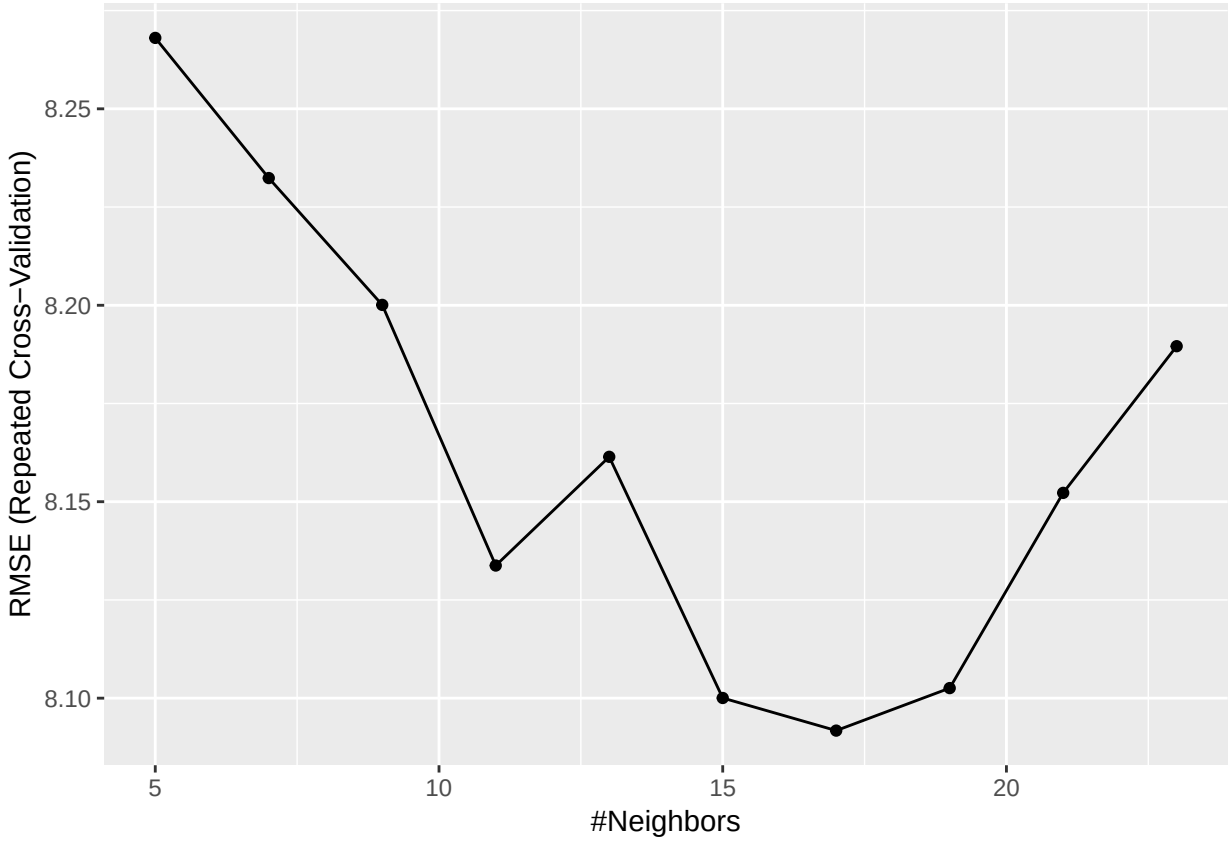
```
## Created from 331 samples and 6 variables
##
## Pre-processing:
##   - centered (6)
##   - ignored (0)
##   - scaled (6)
```

```
set.seed(1)
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 5)
knn_fit <- train(house_price ~., data = train, method = "knn",
```

```
            trControl = trctrl,
            preProcess = c("center", "scale"),
            tuneLength = 10)
knn_fit
```

```
## k-Nearest Neighbors
##
## 331 samples
##   6 predictor
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 296, 299, 298, 298, 298, 298, ...
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   MAE
##    5  8.268049  0.6394073  5.680187
##    7  8.232378  0.6417149  5.762707
##    9  8.200095  0.6441582  5.701154
##   11  8.133758  0.6490844  5.623692
##   13  8.161427  0.6463116  5.621314
##   15  8.100042  0.6519151  5.606818
##   17  8.091746  0.6545951  5.607166
##   19  8.102562  0.6555240  5.615482
##   21  8.152253  0.6533721  5.656182
##   23  8.189581  0.6522569  5.695976
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 17.
```

```
# Plot model error RMSE vs different values of k
ggplot(knn_fit)
```




```
# Best tuning parameter k that minimizes the RMSE
knn_fit$bestTune
```

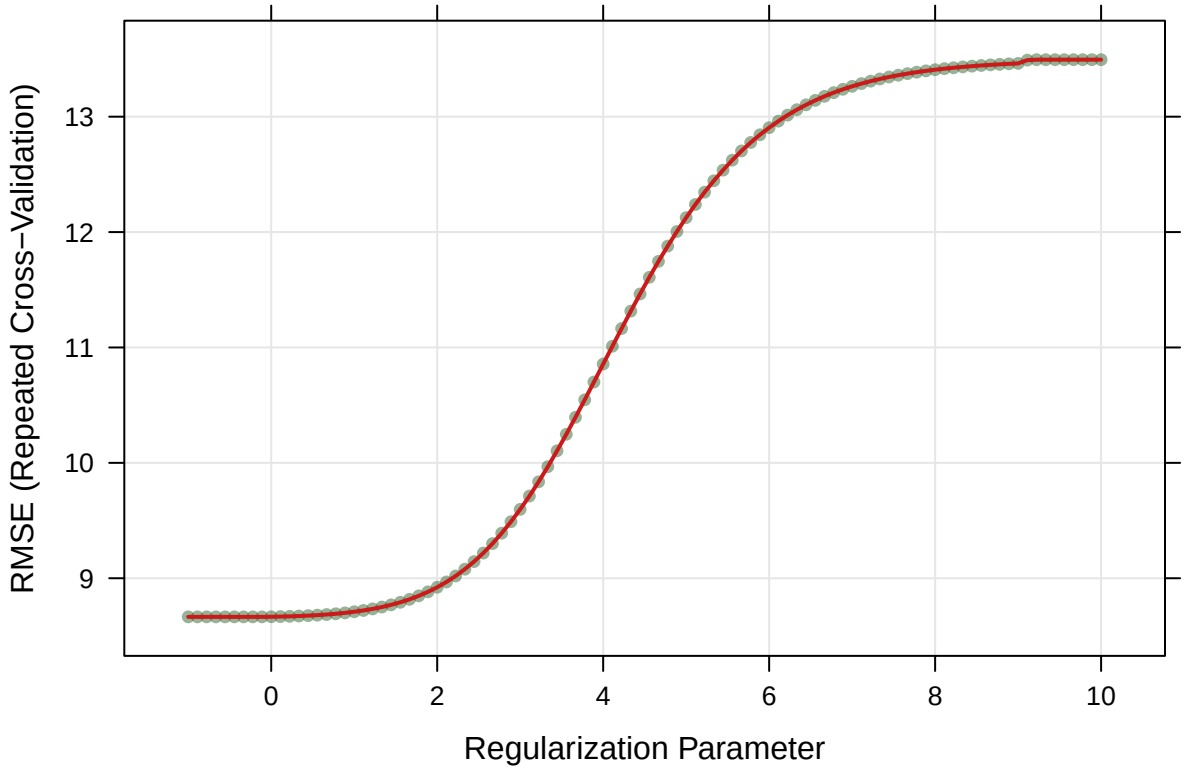
```
##    k
## 7 17
```

```
# Make predictions on the test data
knn_predict <- knn_fit %>% predict(test)
# Compute the prediction error RMSE
RMSE(knn_predict, test$house_price)
```

```
## [1] 8.208905
```

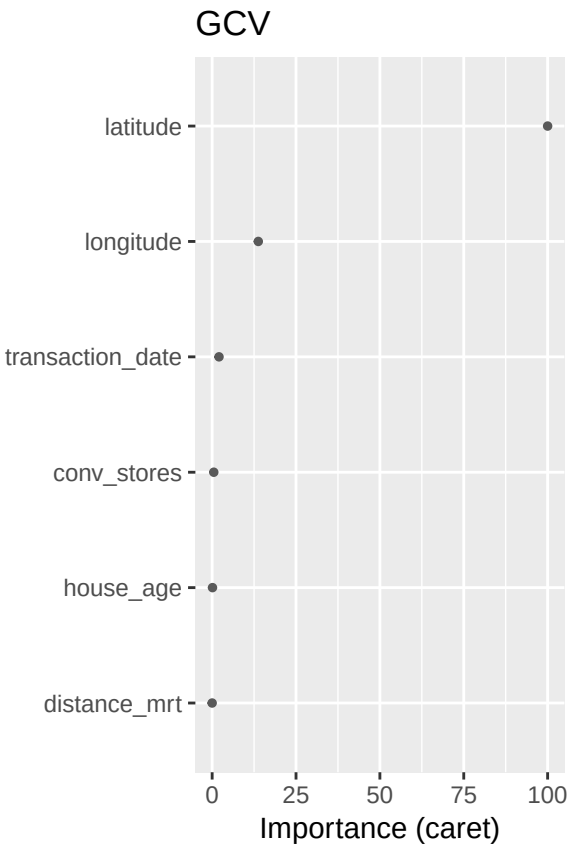
Ridge Regression

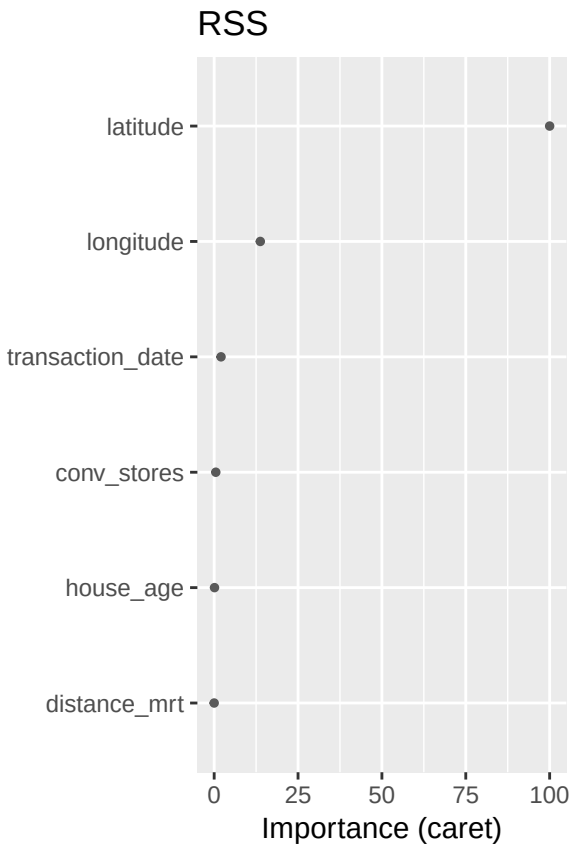
Assumptions:




```
##   alpha    lambda
## 8     0 0.8007374
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)      -1.905169e+04
## transaction_date  4.718759e+00
## house_age        -2.518224e-01
## distance_mrt     -3.783600e-03
## conv_stores       1.124206e+00
## latitude          2.301424e+02
## longitude         3.165906e+01
```



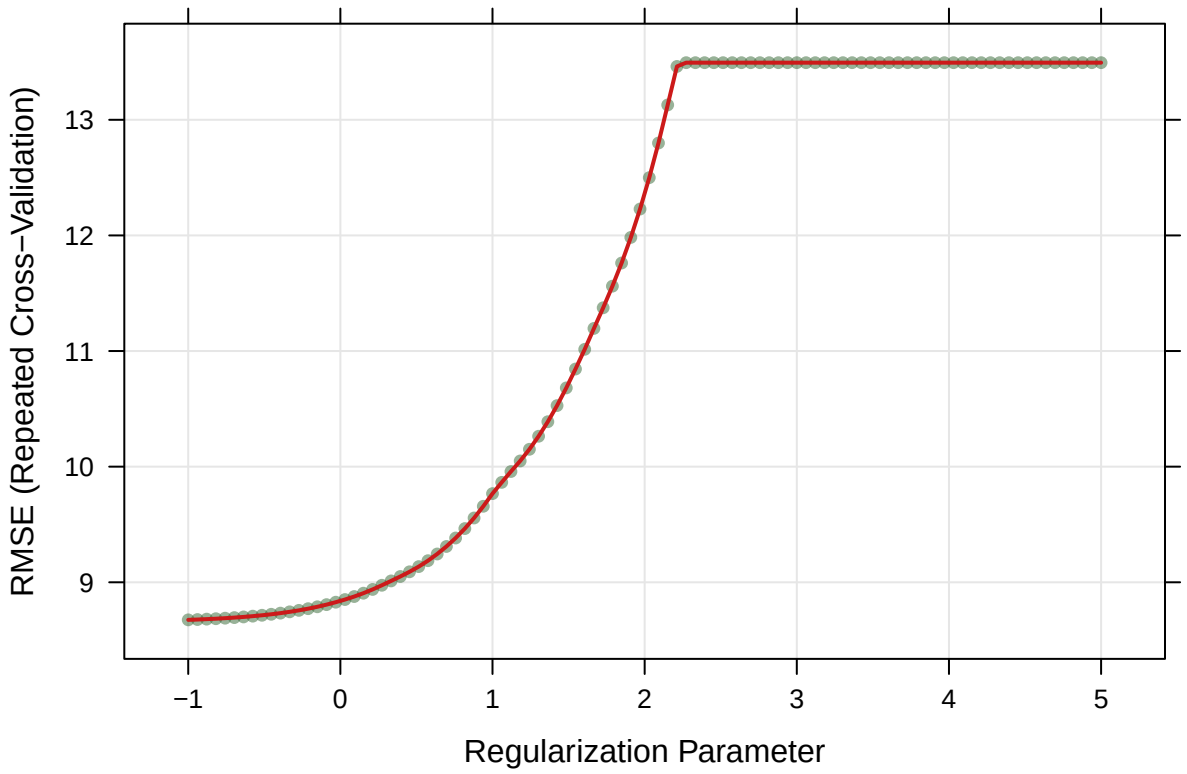


Lasso

Assumptions:

```
set.seed(123)
lasso.fit <- train(x, y, method = "glmnet", tuneGrid = expand.grid(alpha = 1, lambda = exp(seq(-1, 5, l

plot(lasso.fit, xTrans = function(x) log(x))
```




```
lasso.fit$bestTune
```

```
##   alpha    lambda
## 1     1 0.3678794
```

```
coef(lasso.fit$finalModel,lasso.fit$bestTune$lambda)
```

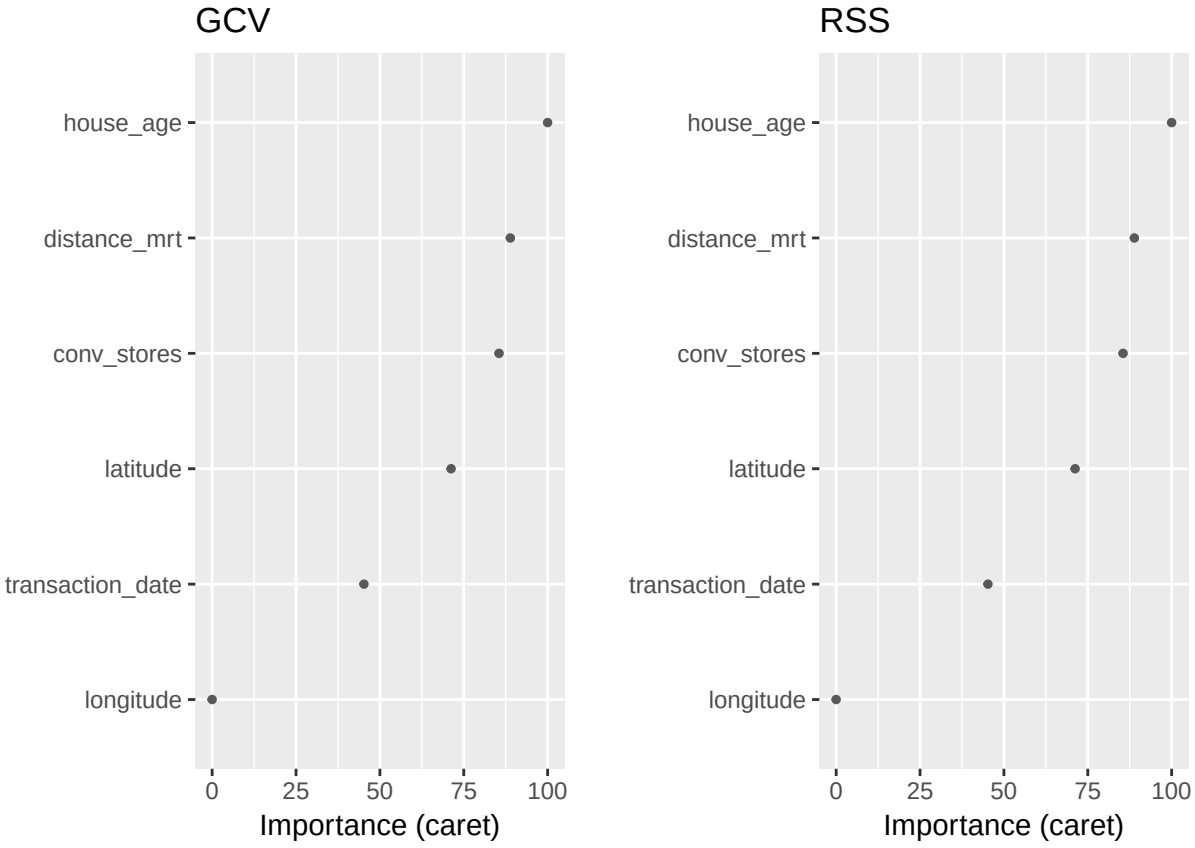
```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                                1
## (Intercept)      -1.288540e+04
## transaction_date  3.811669e+00
## house_age        -2.348535e-01
## distance_mrt     -4.249784e-03
## conv_stores       1.063351e+00
## latitude          2.104339e+02
## longitude         .
```

Linear

Assumptions: Linear relationship, Residuals normality, No or little multicollinearity, No auto-correlation, Homoscedasticity

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.667  -5.412  -0.967   4.217  75.190
##
## Coefficients:
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.444e+04  6.775e+03  -2.132  0.03364 *
## transaction_date  5.149e+00  1.557e+00   3.307  0.00103 **
## house_age        -2.697e-01  3.853e-02  -7.000 1.06e-11 ***
## distance_mrt     -4.488e-03  7.180e-04  -6.250 1.04e-09 ***
## conv_stores       1.133e+00  1.882e-01   6.023 3.83e-09 ***
## latitude          2.255e+02  4.457e+01   5.059 6.38e-07 ***
## longitude        -1.243e+01  4.858e+01  -0.256  0.79820
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.858 on 407 degrees of freedom
## Multiple R-squared:  0.5824, Adjusted R-squared:  0.5762
## F-statistic:  94.6 on 6 and 407 DF,  p-value: < 2.2e-16
```



Conclusion

Based on our results...