SVM Regression

Benjamin Frenkel & Justin Hardy

Load packages

```
library(e1071)
library(MASS)
```

Read in the data set file

```
Data <- read.csv("Housing Price Predictions.csv")

#take only the first 10,000 rows
Data <- Data[1:10000,]
```

Clean data

```
#Factor columns (columns with only a few possible values)
Data$POSTED_BY <- factor(Data$POSTED_BY)
Data$UNDER_CONSTRUCTION <- factor(Data$UNDER_CONSTRUCTION)
Data$BHK_OR_RK <- factor(Data$BHK_OR_RK)
Data$RERA <- factor (Data$RERA)
Data$READY_TO_MOVE <- factor(Data$READY_TO_MOVE)
Data$RESALE <- factor(Data$RESALE)

#Remove useless columns
Data <- subset(Data, select = -c(ADDRESS))

#Rename columns if needed
names(Data) [names(Data) == 'TARGET.PRICE_IN_LACS.'] <- "PRICE_IN_LACS"

#Delete NA rows
Data <- Data[complete.cases(Data),]
```

Divide into train/test/validate

```r
set.seed(1234)
group <- c(train=.6, test=.2, validate=.2)
i <- sample(cut(1:nrow(Data), nrow(Data)*cumsum(c(0, group)), labels=names(group)))

train <- Data[i=="train",]
test <- Data[i=="test",]
vald <- Data[i=="validate",]
```

Data Exploration

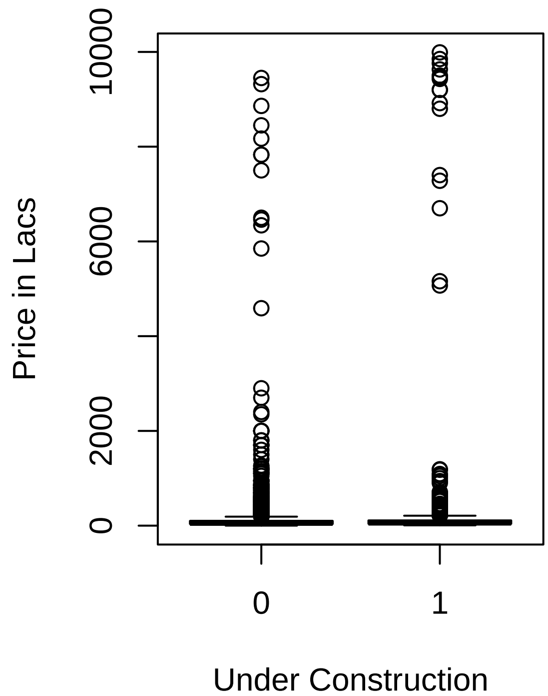
Structure

```r
summary(train)
```

```
##    POSTED_BY    UNDER_CONSTRUCTION RERA        BHK_NO.       BHK_OR_RK
##  Builder: 129   0:4906             0:4063   Min.   : 1.000   BHK:5996
##  Dealer :3732   1:1094             1:1937   1st Qu.: 2.000   RK :   4
##  Owner  :2139                               Median : 2.000
##                                             Mean   : 2.382
##                                             3rd Qu.: 3.000
##                                             Max.   :20.000
##    SQUARE_FT      READY_TO_MOVE RESALE     LONGITUDE         LATITUDE
##  Min.   :     3   0:1094        0: 409   Min.   : 3.161   Min.   :-117.00
##  1st Qu.:   900   1:4906        1:5591   1st Qu.:18.430   1st Qu.:  73.76
##  Median :  1175                          Median :20.264   Median :  77.30
##  Mean   :  1906                          Mean   :21.262   Mean   :  76.73
##  3rd Qu.:  1550                          3rd Qu.:26.901   3rd Qu.:  77.76
##  Max.   :230000                          Max.   :59.913   Max.   : 136.00
##  PRICE_IN_LACS
##  Min.   :   0.85
##  1st Qu.:  37.00
##  Median :  61.00
##  Mean   : 139.64
##  3rd Qu.: 100.00
##  Max.   :9990.00
```

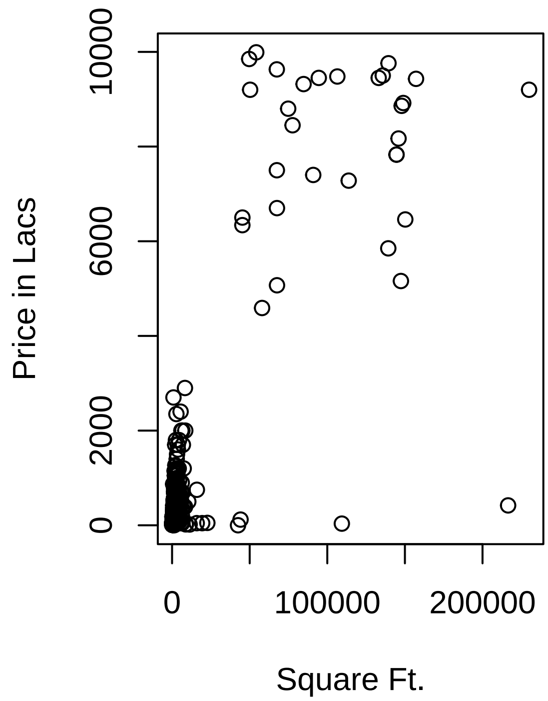
Graphs & Plots

```r
par(mfrow=c(1,2))
plot(train$UNDER_CONSTRUCTION, train$PRICE_IN_LACS, xlab="Under Construction", ylab="Price in Lacs")
```



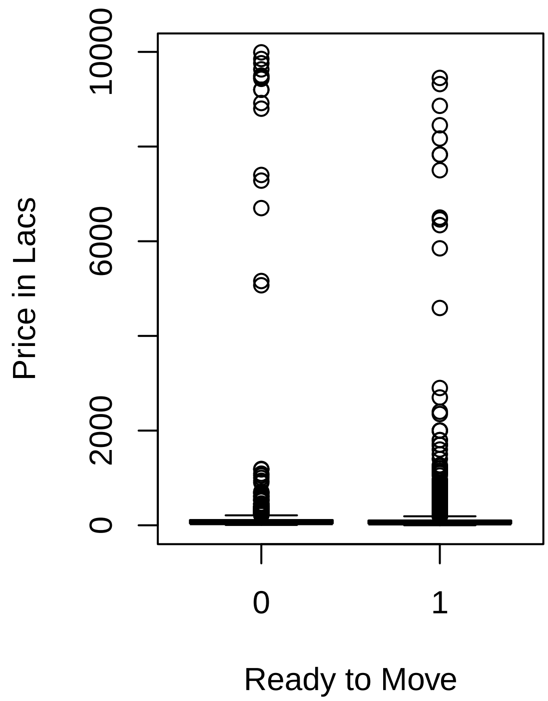
There does not appear to be much of a correlation between whether a house is under construction or not and its price.

```
par(mfrow=c(1,2))
plot(train$SQUARE_FT, train$PRICE_IN_LACS, xlab="Square Ft.", ylab="Price in Lacs")
```



Aside for a few outliers there seems to be a good correlation between the square ft. of the house and its price, generally as square ft. increases price increases.

```
par(mfrow=c(1,2))
plot(train$READY_TO_MOVE, train$PRICE_IN_LACS, xlab="Ready to Move", ylab="Price in Lacs")
```



The factor of whether the house is ready to move into or not seems to have a decent correlation with the price of the house. The house being ready to move into generally correlates to the price being slightly higher than if it was not ready to move into.

Models

Linear Kernel

```
svm1 <- svm(PRICE_IN_LACS~., data=train, kernel="linear", cost=10, scale=TRUE)
summary(svm1)
```

```
##
## Call:
## svm(formula = PRICE_IN_LACS ~ ., data = train, kernel = "linear",
##     cost = 10, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  linear
##        cost:  10
##       gamma:  0.08333333
##     epsilon:  0.1
```

```
##
##
## Number of Support Vectors:  1372
```

```r
pred <- predict(svm1, newdata=test)
cor_svm1 <- cor(pred, test$PRICE_IN_LACS)
mse_svm1 <- mean((pred - test$PRICE_IN_LACS)^2)

cat(paste("Correlation: ", cor_svm1), paste("MSE: ", mse_svm1), sep='\n')
```

```
## Correlation:  0.879301336888865
## MSE:  129331.027795529
```

Tune

```r
tune_svm1 <- tune(svm, PRICE_IN_LACS~., data=vald, kernel="linear",
                  ranges=list(cost=c(0.001, 0.01, 0.1, 1, 5, 10, 100)))
summary(tune_svm1)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##   cost
##  0.001
##
## - best performance: 377222.5
##
## - Detailed performance results:
##    cost    error dispersion
## 1 1e-03 377222.5   341286.8
## 2 1e-02 526419.5   824425.0
## 3 1e-01 499119.5  1206252.0
## 4 1e+00 502286.3  1243661.2
## 5 5e+00 503594.8  1247823.9
## 6 1e+01 503592.1  1247804.0
## 7 1e+02 503577.2  1247755.4
```

Evaluate on best linear svm

```r
pred <- predict(tune_svm1$best.model, newdata=test)
cor_svm1_tune <- cor(pred, test$PRICE_IN_LACS)
mse_svm1_tune <- mean((pred - test$PRICE_IN_LACS)^2)
```

Try a polynomial kernel

```r
svm2 <- svm(PRICE_IN_LACS~., data=train, kernel="polynomial", cost=10, scale=TRUE)
summary(svm2)
```

```
##
## Call:
## svm(formula = PRICE_IN_LACS ~ ., data = train, kernel = "polynomial",
##     cost = 10, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  3
##       gamma:  0.08333333
##      coef.0:  0
##     epsilon:  0.1
##
##
## Number of Support Vectors:  1246
```

```r
pred <- predict(svm2, newdata=test)
cor_svm2 <- cor(pred, test$PRICE_IN_LACS)
mse_svm2 <- mean((pred - test$PRICE_IN_LACS)^2)

cat(paste("Correlation: ", cor_svm2), paste("MSE: ", mse_svm2), sep='\n')
```

```
## Correlation:  0.902228652577206
## MSE:  85471.0640896984
```

Try a radial kernel

```r
svm3 <- svm(PRICE_IN_LACS~., data=train, kernel="radial", cost=10, gamma=1, scale=TRUE)
summary(svm3)
```

```
##
## Call:
## svm(formula = PRICE_IN_LACS ~ ., data = train, kernel = "radial",
##     cost = 10, gamma = 1, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  10
##       gamma:  1
##     epsilon:  0.1
```

```
##
##
## Number of Support Vectors:  1200
```

```r
pred <- predict(svm3, newdata=test)
cor_svm3 <- cor(pred, test$PRICE_IN_LACS)
mse_svm3 <- mean((pred - test$PRICE_IN_LACS)^2)

cat(paste("Correlation: ", cor_svm3), paste("MSE: ", mse_svm3), sep='\n')
```

```
## Correlation:  0.75530717567311
## MSE:  209722.955705968
```

Tune hyperperameters

```r
set.seed(1234)
tune.out <- tune(svm, PRICE_IN_LACS~., data=vald, kernel="radial",
                 ranges=list(cost=c(0.1,1,10,100,1000),
                             gamma=c(0.5,1,2,3,4)))
summary(tune.out)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##  1000   0.5
##
## - best performance: 187390.8
##
## - Detailed performance results:
##     cost gamma    error dispersion
## 1  1e-01   0.5 373556.8   389446.3
## 2  1e+00   0.5 341958.9   363517.1
## 3  1e+01   0.5 204143.8   214245.1
## 4  1e+02   0.5 190311.0   196361.4
## 5  1e+03   0.5 187390.8   183937.4
## 6  1e-01   1.0 378515.1   392054.3
## 7  1e+00   1.0 358618.4   378400.1
## 8  1e+01   1.0 291413.7   306713.3
## 9  1e+02   1.0 273653.4   281751.1
## 10 1e+03   1.0 274202.5   275273.3
## 11 1e-01   2.0 381736.5   392927.6
## 12 1e+00   2.0 368593.9   385462.8
## 13 1e+01   2.0 339710.0   359668.8
## 14 1e+02   2.0 330162.8   346294.4
## 15 1e+03   2.0 333547.3   337229.7
## 16 1e-01   3.0 382377.2   393253.7
## 17 1e+00   3.0 370959.5   386733.0
```

```
## 18 1e+01   3.0 350083.2   369655.2
## 19 1e+02   3.0 343299.4   359505.4
## 20 1e+03   3.0 346687.0   351629.1
## 21 1e-01   4.0 382688.8   393397.2
## 22 1e+00   4.0 371793.8   387274.5
## 23 1e+01   4.0 353891.1   372166.9
## 24 1e+02   4.0 347888.2   363878.2
## 25 1e+03   4.0 353012.7   359620.4
```

```
svm4 <- svm(PRICE_IN_LACS~., data=train, kernel="radial", cost=100, gamma=0.5, scale=TRUE)
summary(svm4)
```

```
##
## Call:
## svm(formula = PRICE_IN_LACS ~ ., data = train, kernel = "radial",
##     cost = 100, gamma = 0.5, scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  100
##       gamma:  0.5
##     epsilon:  0.1
##
##
## Number of Support Vectors:  1138
```

```
pred <- predict(svm4, newdata=test)
cor_svm4 <- cor(pred, test$PRICE_IN_LACS)
mse_svm4 <- mean((pred - test$PRICE_IN_LACS)^2)
```

Analysis

In this regression section of the assignment, three SVM regressions were performed on the data. Linear kernel, polynomial kernel, and radial kernel. Of the three types of SVM regression svm2 (Polynomial kernel) had the highest correlation. This is likely due to the data not being linearly separable, so the linear kernel is not the best choice of model for the data.

Now, between linear kernel and radial kernel, linear kernel has a much higher correlation than radial kernel. So, while polynomial has the best correlation, linear kernel is clearly second best, and radial kernel is the worst.

The mean squared error lines up with the correlation as polynomial kernel has the lowest MSE, followed by linear kernel, and with radial kernel having the highest MSE.