Regression

Vladimir Sokolov

10/11/2022

This notebook explores King County House Sales data from Kaggle.

Load the kc_housing_data.csv file and change waterfront into a factor.

```
df <- read.csv("kc_house_data.csv")
df$waterfront <- factor(df$waterfront)
str(df)
```

```
## 'data.frame':    21613 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

Simplify variables.

```
df<-df[-c(1,2,8,14:21)]
str(df)
```

```
## 'data.frame':    21613 obs. of  10 variables:
##  $ price      : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms   : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms  : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living: int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
```

```
##  $ sqft_lot   : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ waterfront : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ view       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition  : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade      : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
```

Check for null values.

```
sapply(df, function(x) sum(is.na(x)))
```

```
##       price    bedrooms   bathrooms sqft_living    sqft_lot  waterfront 
##           0           0           0           0           0           0 
##        view   condition       grade  sqft_above 
##           0           0           0           0
```

#A.Divide into 80/20 train/test.

```
set.seed(12345)
i <- sample(1:nrow(df), nrow(df)*.8, replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

#B.Explore data.

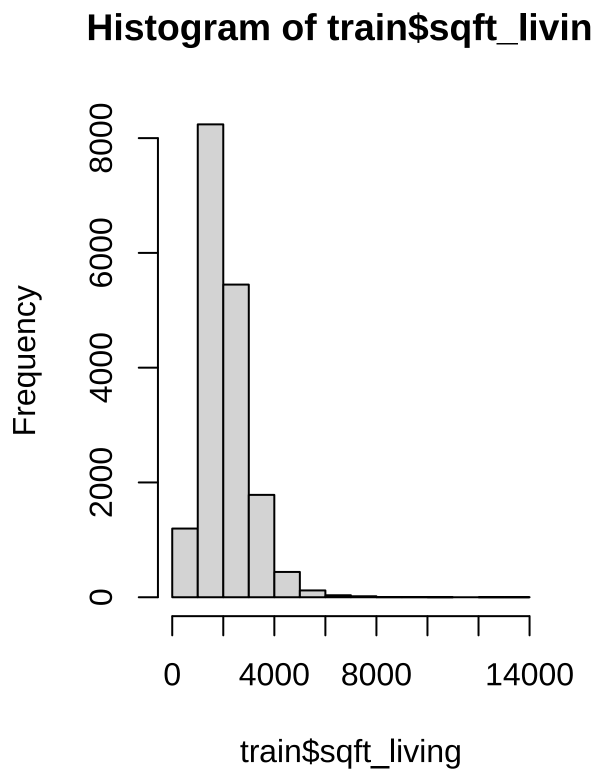
```
summary(train$sqft_living)
```

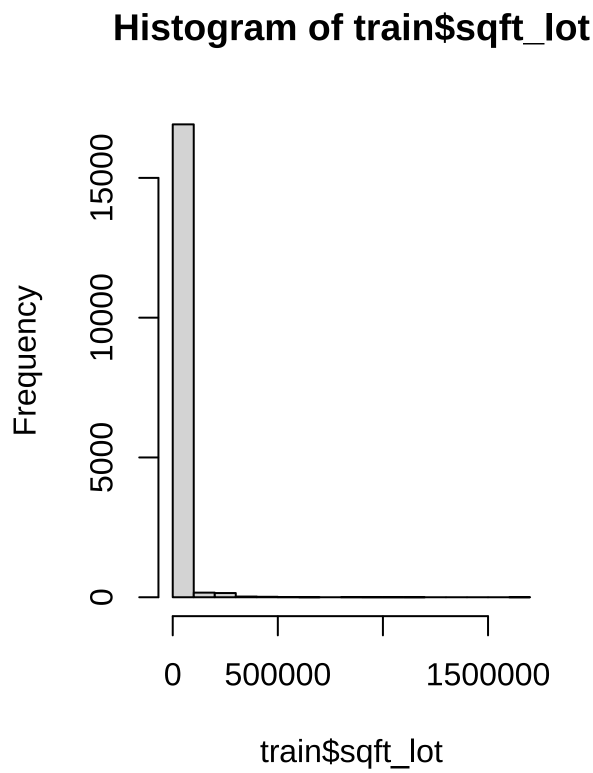
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     290    1428    1900    2075    2530   13540
```

```
summary(train$sqft_lot)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     572    5040    7620   15152   10660 1651359
```

```
par(mfrow=c(1,2))
hist(train$sqft_living)
hist(train$sqft_lot)
```





#C.Regressions Linear Regression

```
lm1 <- lm(price ~., data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = price ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1179796  -123990   -16225    94428  4561152
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.872e+05  1.933e+04 -35.547  < 2e-16 ***
## bedrooms    -3.585e+04  2.403e+03 -14.918  < 2e-16 ***
## bathrooms   -1.846e+04  3.652e+03  -5.056 4.33e-07 ***
## sqft_living  2.262e+02  5.039e+00  44.894  < 2e-16 ***
## sqft_lot    -3.199e-01  4.275e-02  -7.481 7.71e-14 ***
## waterfront1  5.825e+05  2.276e+04  25.594  < 2e-16 ***
## view         6.056e+04  2.678e+03  22.609  < 2e-16 ***
## condition    5.471e+04  2.826e+03  19.364  < 2e-16 ***
## grade        9.982e+04  2.538e+03  39.333  < 2e-16 ***
## sqft_above  -2.642e+01  4.818e+00  -5.484 4.21e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 231900 on 17280 degrees of freedom
## Multiple R-squared:  0.6099, Adjusted R-squared:  0.6097
## F-statistic:  3001 on 9 and 17280 DF,  p-value: < 2.2e-16
```

```
pred1 <- predict(lm1, newdata = test)
cor_lm1 <- cor(pred1, test$price)
mse_lm1 <- mean((pred1-test$price)^2)
print(paste("cor1=",cor_lm1))
```

```
## [1] "cor1= 0.766390552599098"
```

```
print(paste("mse1=",mse_lm1))
```

```
## [1] "mse1= 51099881423.065"
```

kNN Regression

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
train$waterfront <- as.integer(train$waterfront)
test$waterfront <- as.integer(test$waterfront)
fit <- knnreg(train[,2:9], train[,1],k=3)
pred2 <- predict(fit, test[2:9])
cor_knn1 <- cor(pred2, test$price)
mse_knn1 <- mean((pred2 - test$price)^2)
print(paste("cor2=", cor_knn1))
```

```
## [1] "cor2= 0.618320429774807"
```

```
print(paste("mse2=", mse_knn1))
```

```
## [1] "mse2= 80780449918.8132"
```

Decision tree regression

```
library(tree)
tree1 <- tree(price ~., data=train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = price ~ ., data = train)
## Variables actually used in tree construction:
```

```
## [1] "grade"       "sqft_living" "waterfront"
## Number of terminal nodes:  7
## Residual mean deviance:  5.956e+10 = 1.029e+15 / 17280
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2801000  -138300   -33340        0    99040  2931000
```

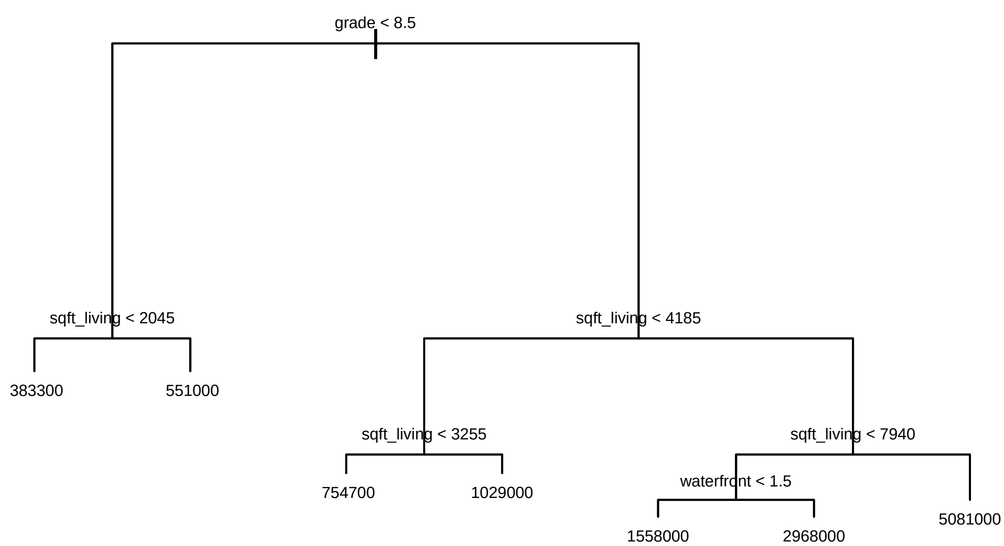
```r
pred3 <- predict(tree1, newdata=test)
cor_tree <- cor(pred3, test$price)
mse_tree <- mean((pred3 - test$price)^2)
print(paste("cor3=", cor_tree))
```

```
## [1] "cor3= 0.6950164475377"
```

```r
print(paste("mse3=", mse_tree))
```

```
## [1] "mse3= 63805681110.7403"
```

```r
plot(tree1)
text(tree1, cex=0.5, pretty=0)
```



Comparison

```r
print(paste("corlr=",cor_lm1))
```

```
## [1] "corlr= 0.766390552599098"
```

```r
print(paste("corknn=", cor_knn1))
```

```
## [1] "corknn= 0.618320429774807"
```

```r
print(paste("cortree=", cor_tree))
```

```
## [1] "cortree= 0.6950164475377"
```

#D.Analysis Linear regression did the best followed by Decision tree then knn. Outliers and not easily splittable data made knn and decision trees perform worse than the standard linear regression.