Machine Learning Project

Michal Heydel

29 November 2017

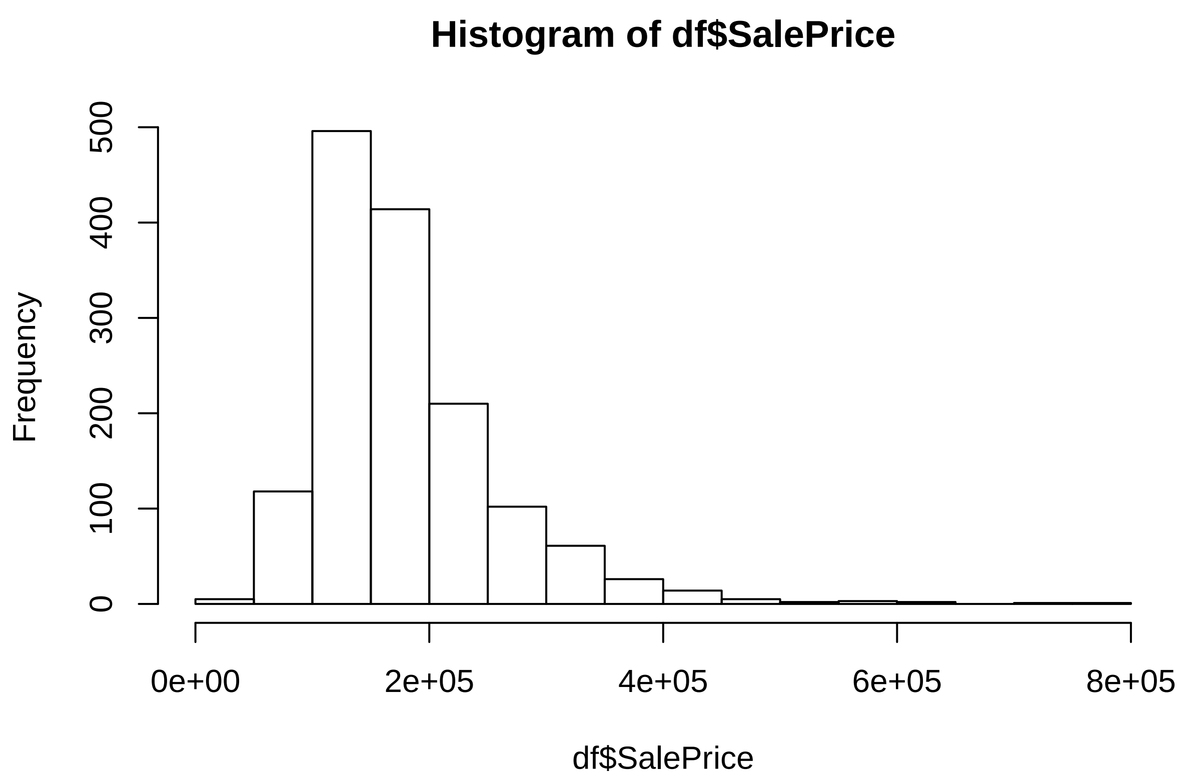
```
setwd("C:/Users/Michal/Documents/01- Master Degree/GitHub/ST443-Project-group9/Housing price data")
getwd()
```

```
## [1] "C:/Users/Michal/Documents/01- Master Degree/GitHub/ST443-Project-group9/Housing price data"
```

```
train = read.csv("train.csv", row.names = "Id", stringsAsFactors=FALSE)
testing_kaggle = read.csv("test.csv", row.names = "Id", stringsAsFactors=FALSE)
```

```
#combining train and test data for quicker data prep
testing_kaggle$SalePrice <- NA
train$isTrain <- 1
testing_kaggle$isTrain <- 0
df <- rbind(train,testing_kaggle)
```

```
hist(df$SalePrice)
```




```
colSums(sapply(df, is.na))
```

```
##    MSSubClass      MSZoning   LotFrontage       LotArea        Street
##             0             4           486             0             0
##         Alley      LotShape   LandContour     Utilities     LotConfig
```

```
##          2721             0             0             2             0
##     LandSlope  Neighborhood    Condition1    Condition2      BldgType
##             0             0             0             0             0
##    HouseStyle   OverallQual   OverallCond     YearBuilt  YearRemodAdd
##             0             0             0             0             0
##     RoofStyle      RoofMatl   Exterior1st   Exterior2nd    MasVnrType
##             0             0             1             1            24
##    MasVnrArea     ExterQual     ExterCond    Foundation      BsmtQual
##            23             0             0             0            81
##      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1  BsmtFinType2
##            82            82            79             1            80
##    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating     HeatingQC
##             1             1             1             0             0
##    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF  LowQualFinSF
##             0             1             0             0             0
##     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath      HalfBath
##             0             2             2             0             0
##  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd    Functional
##             0             0             1             0             2
##    Fireplaces   FireplaceQu    GarageType   GarageYrBlt  GarageFinish
##             0          1420           157           159           159
##    GarageCars    GarageArea    GarageQual    GarageCond    PavedDrive
##             1             1           159           159             0
##    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch   ScreenPorch
##             0             0             0             0             0
##      PoolArea        PoolQC         Fence   MiscFeature       MiscVal
##             0          2909          2348          2814             0
##        MoSold        YrSold      SaleType SaleCondition     SalePrice
##             0             0             1             0          1459
##       isTrain
##             0
```

```r
for(i in colnames(df[,sapply(df, is.character)])){

  df[,i][which(is.na(df[,i]))] <- "None"
}
```

```r
colSums(sapply(df, is.na))
```

```
##    MSSubClass      MSZoning   LotFrontage       LotArea        Street
##             0             0           486             0             0
##         Alley      LotShape   LandContour     Utilities     LotConfig
##             0             0             0             0             0
##     LandSlope  Neighborhood    Condition1    Condition2      BldgType
##             0             0             0             0             0
##    HouseStyle   OverallQual   OverallCond     YearBuilt  YearRemodAdd
##             0             0             0             0             0
##     RoofStyle      RoofMatl   Exterior1st   Exterior2nd    MasVnrType
##             0             0             0             0             0
##    MasVnrArea     ExterQual     ExterCond    Foundation      BsmtQual
##            23             0             0             0             0
##      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1  BsmtFinType2
##             0             0             0             1             0
##    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating     HeatingQC
##             1             1             1             0             0
```

```
##    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF  LowQualFinSF
##             0             0             0             0             0
##     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath      HalfBath
##             0             2             2             0             0
##  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd    Functional
##             0             0             0             0             0
##    Fireplaces   FireplaceQu    GarageType   GarageYrBlt  GarageFinish
##             0             0             0           159             0
##    GarageCars    GarageArea    GarageQual    GarageCond    PavedDrive
##             1             1             0             0             0
##    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch   ScreenPorch
##             0             0             0             0             0
##      PoolArea        PoolQC         Fence   MiscFeature       MiscVal
##             0             0             0             0             0
##        MoSold        YrSold      SaleType SaleCondition     SalePrice
##             0             0             0             0          1459
##       isTrain
##             0
```

```r
df$LotFrontage[which(is.na(df$LotFrontage))] <- median(df$LotFrontage,na.rm = T)
```

```r
df$MasVnrArea[which(is.na(df$MasVnrArea))] <- mean(df$LotFrontage,na.rm = T)
```

```r
x = c("BsmtFinSF1","BsmtFinSF2", "BsmtUnfSF", "TotalBsmtSF", "BsmtFullBath", "BsmtHalfBath", "GarageYrBl

for(i in x){

  df[,i][which(is.na(df[,i]))] <- 0
}
```

```r
colSums(sapply(df, is.na))
```

```
##    MSSubClass      MSZoning   LotFrontage       LotArea        Street
##             0             0             0             0             0
##         Alley      LotShape   LandContour     Utilities     LotConfig
##             0             0             0             0             0
##     LandSlope  Neighborhood    Condition1    Condition2      BldgType
##             0             0             0             0             0
##    HouseStyle   OverallQual   OverallCond     YearBuilt  YearRemodAdd
##             0             0             0             0             0
##     RoofStyle      RoofMatl   Exterior1st   Exterior2nd    MasVnrType
##             0             0             0             0             0
##    MasVnrArea     ExterQual     ExterCond    Foundation      BsmtQual
##             0             0             0             0             0
##      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1  BsmtFinType2
##             0             0             0             0             0
##    BsmtFinSF2     BsmtUnfSF   TotalBsmtSF       Heating     HeatingQC
##             0             0             0             0             0
##    CentralAir    Electrical     X1stFlrSF     X2ndFlrSF  LowQualFinSF
##             0             0             0             0             0
##     GrLivArea  BsmtFullBath  BsmtHalfBath      FullBath      HalfBath
##             0             0             0             0             0
##  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd    Functional
##             0             0             0             0             0
##    Fireplaces   FireplaceQu    GarageType   GarageYrBlt  GarageFinish
```

```
##              0             0             0             0             0
##     GarageCars    GarageArea    GarageQual    GarageCond    PavedDrive
##              0             0             0             0             0
##     WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch   ScreenPorch
##              0             0             0             0             0
##       PoolArea        PoolQC         Fence   MiscFeature       MiscVal
##              0             0             0             0             0
##         MoSold        YrSold      SaleType SaleCondition     SalePrice
##              0             0             0             0          1459
##        isTrain
##              0
```

```r
for(i in colnames(df[,sapply(df, is.character)])){
    df[,i] <- as.factor(df[,i])
}
```

```r
# These are also categorical Variables
df$MSSubClass <- as.factor(df$MSSubClass)
df$OverallCond <- as.factor(df$OverallCond)
df$OverallQual <- as.factor(df$OverallQual)
```

```r
str(df)
```

```
## 'data.frame':    2919 obs. of  81 variables:
##  $ MSSubClass   : Factor w/ 16 levels "20","30","40",..: 6 1 6 7 6 5 1 6 5 16 ...
##  $ MSZoning     : Factor w/ 6 levels "C (all)","FV",..: 5 5 5 5 5 5 5 5 6 5 ...
##  $ LotFrontage  : int  65 80 68 60 84 85 75 68 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley        : Factor w/ 3 levels "Grvl","None",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
##  $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Utilities    : Factor w/ 3 levels "AllPub","None",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...
##  $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...
##  $ Condition2   : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...
##  $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ HouseStyle   : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3 6 1 2 ...
##  $ OverallQual  : Factor w/ 10 levels "1","2","3","4",..: 7 6 7 7 8 5 8 7 7 5 ...
##  $ OverallCond  : Factor w/ 9 levels "1","2","3","4",..: 5 8 5 5 5 5 5 6 5 6 ...
##  $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
##  $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
##  $ RoofStyle    : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ RoofMatl     : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Exterior1st  : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 15 14 14 14 7 4 9 ...
##  $ Exterior2nd  : Factor w/ 17 levels "AsbShng","AsphShn",..: 15 9 15 17 15 15 15 7 17 9 ...
##  $ MasVnrType   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4 4 3 3 ...
##  $ MasVnrArea   : num  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4 4 ...
##  $ ExterCond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation   : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...
##  $ BsmtQual     : Factor w/ 5 levels "Ex","Fa","Gd",..: 3 3 3 5 3 3 1 3 5 5 ...
##  $ BsmtCond     : Factor w/ 5 levels "Fa","Gd","None",..: 5 5 5 2 5 5 5 5 5 5 ...
```

```
##  $ BsmtExposure : Factor w/ 5 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4 4 ...
##  $ BsmtFinType1 : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1 7 3 ...
##  $ BsmtFinSF1   : num  706 978 486 216 655 ...
##  $ BsmtFinType2 : Factor w/ 7 levels "ALQ","BLQ","GLQ",..: 7 7 7 7 7 7 7 2 7 7 ...
##  $ BsmtFinSF2   : num  0 0 0 0 0 0 0 32 0 0 ...
##  $ BsmtUnfSF    : num  150 284 434 540 490 64 317 216 952 140 ...
##  $ TotalBsmtSF  : num  856 1262 920 756 1145 ...
##  $ Heating      : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical   : Factor w/ 6 levels "FuseA","FuseF",..: 6 6 6 6 6 6 6 6 2 6 ...
##  $ X1stFlrSF    : int  856 1262 920 961 1145 796 1694 1107 1022 1077 ...
##  $ X2ndFlrSF    : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...
##  $ BsmtFullBath : num  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath     : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr : int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 2 2 ...
##  $ KitchenQual  : Factor w/ 5 levels "Ex","Fa","Gd",..: 3 5 3 3 3 5 3 5 5 5 ...
##  $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
##  $ Functional   : Factor w/ 8 levels "Maj1","Maj2",..: 8 8 8 8 8 8 8 8 3 8 ...
##  $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
##  $ FireplaceQu  : Factor w/ 6 levels "Ex","Fa","Gd",..: 4 6 6 3 6 4 3 6 6 6 ...
##  $ GarageType   : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...
##  $ GarageYrBlt  : num  2003 1976 2001 1998 2000 ...
##  $ GarageFinish : Factor w/ 4 levels "Fin","None","RFn",..: 3 3 3 4 3 4 3 3 4 3 ...
##  $ GarageCars   : num  2 2 2 3 3 2 2 2 2 1 ...
##  $ GarageArea   : num  548 460 608 642 836 480 636 484 468 205 ...
##  $ GarageQual   : Factor w/ 6 levels "Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 2 3 ...
##  $ GarageCond   : Factor w/ 6 levels "Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF   : int  0 298 0 0 192 40 255 235 90 0 ...
##  $ OpenPorchSF  : int  61 0 42 35 84 30 57 204 0 4 ...
##  $ EnclosedPorch: int  0 0 0 272 0 0 0 228 205 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 320 0 0 0 0 ...
##  $ ScreenPorch  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : Factor w/ 4 levels "Ex","Fa","Gd",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Fence        : Factor w/ 5 levels "GdPrv","GdWo",..: 5 5 5 5 5 3 5 5 5 5 ...
##  $ MiscFeature  : Factor w/ 5 levels "Gar2","None",..: 2 2 2 2 2 4 2 4 2 2 ...
##  $ MiscVal      : int  0 0 0 0 0 700 0 350 0 0 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...
##  $ SaleType     : Factor w/ 10 levels "COD","Con","ConLD",..: 10 10 10 10 10 10 10 10 10 10 ...
##  $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...
##  $ SalePrice    : int  208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...
##  $ isTrain      : num  1 1 1 1 1 1 1 1 1 1 ...
```

REGRESSION

```r
train <- df[df$isTrain==1,]
test <- df[df$isTrain==0,]

train$isTrain <- NULL
```

```r
smp_size = floor(0.8 * nrow(train))
set.seed(1)

train_ind <-sample(seq_len(nrow(train)),smp_size, replace = F)
```

```r
library(boot)
library(leaps)
```

```r
# FROM CLASS 5
K <- 10
set.seed(11)
folds <-sample(rep(1:10, length=nrow(train)))
table(folds)
```

```
## folds
##   1   2   3   4   5   6   7   8   9  10
## 146 146 146 146 146 146 146 146 146 146
```

```r
## We initialize a error matrix with row (10 different folds) and column (19 different predictors)
cv.errors <-matrix(0, 10, 19)
```

Below is copied from a link, don't use it. Need to write alone a code.

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:boot':
##
##     melanoma

## Loading required package: ggplot2
```

Experimenting with Machine Learning Algorithms

Model 1: Linear Model

```r
myControl = trainControl(method = "cv", number = 5, verboseIter = FALSE)
model_lm = train(SalePrice ~ .,
              data = train,
              method = "lm",
              trControl = myControl)
model_lm
```

```
## Linear Regression
##
```

```
## 1460 samples
##   79 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1168, 1168, 1168, 1168, 1168
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   53985.31  0.6465696  20403.24
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Model 2: Random Forest

```r
model_rf = train(SalePrice ~ .,
              data = train,
              tuneLength = 1,
              method = "ranger",
              importance = 'impurity',
              trControl = myControl)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
model_rf
```

```
## Random Forest
##
## 1460 samples
##   79 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1168, 1168, 1168, 1167, 1169
## Resampling results across tuning parameters:
##
##   splitrule   RMSE      Rsquared   MAE
##   variance    30600.03  0.8670470  17607.06
##   extratrees  32164.47  0.8542113  18741.26
##
## Tuning parameter 'mtry' was held constant at a value of 17
## RMSE was used to select the optimal model using  the smallest value.
## The final values used for the model were mtry = 17 and splitrule
##  = variance.
```

Model 3: Random Forest with two mtry values

```
model_rf2 = train(SalePrice ~ .,
                  data = train,
                  tuneLength = 2,
                  method = "ranger",
                  importance = 'impurity',
                  trControl = myControl)
model_rf2
```

```
## Random Forest
##
## 1460 samples
##   79 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1168, 1168, 1168, 1169, 1167
## Resampling results across tuning parameters:
##
##   mtry  splitrule   RMSE      Rsquared   MAE
##     2   variance    49050.05  0.7741978  31242.40
##     2   extratrees  52188.52  0.7353499  33989.20
##   296   variance    30451.39  0.8527951  18072.11
##   296   extratrees  30461.05  0.8581483  17855.25
##
## RMSE was used to select the optimal model using  the smallest value.
## The final values used for the model were mtry = 296 and splitrule
##  = variance.
```

```
fit.glmnet <- train(SalePrice~.,train,trControl = myControl,
                    method="glmnet",tuneGrid=expand.grid(.alpha = seq(0,1,by=0.05),
                                                         .lambda = seq(0, 0.08, by = 0.01)))
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-13
```

```
print(fit.glmnet)
```

```
## glmnet
##
## 1460 samples
##   79 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1168, 1169, 1168, 1167, 1168
## Resampling results across tuning parameters:
##
##   alpha  lambda  RMSE      Rsquared   MAE
##   0.00   0.00    34525.97  0.8120356  18512.34
##   0.00   0.01    34525.97  0.8120356  18512.34
##   0.00   0.02    34525.97  0.8120356  18512.34
```

```
##    0.00   0.03    34525.97  0.8120356  18512.34
##    0.00   0.04    34525.97  0.8120356  18512.34
##    0.00   0.05    34525.97  0.8120356  18512.34
##    0.00   0.06    34525.97  0.8120356  18512.34
##    0.00   0.07    34525.97  0.8120356  18512.34
##    0.00   0.08    34525.97  0.8120356  18512.34
##    0.05   0.00    38654.97  0.7761076  18576.35
##    0.05   0.01    38654.97  0.7761076  18576.35
##    0.05   0.02    38654.97  0.7761076  18576.35
##    0.05   0.03    38654.97  0.7761076  18576.35
##    0.05   0.04    38654.97  0.7761076  18576.35
##    0.05   0.05    38654.97  0.7761076  18576.35
##    0.05   0.06    38654.97  0.7761076  18576.35
##    0.05   0.07    38654.97  0.7761076  18576.35
##    0.05   0.08    38654.97  0.7761076  18576.35
##    0.10   0.00    39036.23  0.7730362  18630.90
##    0.10   0.01    39036.23  0.7730362  18630.90
##    0.10   0.02    39036.23  0.7730362  18630.90
##    0.10   0.03    39036.23  0.7730362  18630.90
##    0.10   0.04    39036.23  0.7730362  18630.90
##    0.10   0.05    39036.23  0.7730362  18630.90
##    0.10   0.06    39036.23  0.7730362  18630.90
##    0.10   0.07    39036.23  0.7730362  18630.90
##    0.10   0.08    39036.23  0.7730362  18630.90
##    0.15   0.00    38994.97  0.7736528  18661.74
##    0.15   0.01    38994.97  0.7736528  18661.74
##    0.15   0.02    38994.97  0.7736528  18661.74
##    0.15   0.03    38994.97  0.7736528  18661.74
##    0.15   0.04    38994.97  0.7736528  18661.74
##    0.15   0.05    38994.97  0.7736528  18661.74
##    0.15   0.06    38994.97  0.7736528  18661.74
##    0.15   0.07    38994.97  0.7736528  18661.74
##    0.15   0.08    38994.97  0.7736528  18661.74
##    0.20   0.00    39806.17  0.7663606  18679.11
##    0.20   0.01    39806.17  0.7663606  18679.11
##    0.20   0.02    39806.17  0.7663606  18679.11
##    0.20   0.03    39806.17  0.7663606  18679.11
##    0.20   0.04    39806.17  0.7663606  18679.11
##    0.20   0.05    39806.17  0.7663606  18679.11
##    0.20   0.06    39806.17  0.7663606  18679.11
##    0.20   0.07    39806.17  0.7663606  18679.11
##    0.20   0.08    39806.17  0.7663606  18679.11
##    0.25   0.00    39823.83  0.7662976  18675.63
##    0.25   0.01    39823.83  0.7662976  18675.63
##    0.25   0.02    39823.83  0.7662976  18675.63
##    0.25   0.03    39823.83  0.7662976  18675.63
##    0.25   0.04    39823.83  0.7662976  18675.63
##    0.25   0.05    39823.83  0.7662976  18675.63
##    0.25   0.06    39823.83  0.7662976  18675.63
##    0.25   0.07    39823.83  0.7662976  18675.63
##    0.25   0.08    39823.83  0.7662976  18675.63
##    0.30   0.00    40092.88  0.7639935  18684.92
##    0.30   0.01    40092.88  0.7639935  18684.92
##    0.30   0.02    40092.88  0.7639935  18684.92
```

```
##    0.30   0.03    40092.88  0.7639935  18684.92
##    0.30   0.04    40092.88  0.7639935  18684.92
##    0.30   0.05    40092.88  0.7639935  18684.92
##    0.30   0.06    40092.88  0.7639935  18684.92
##    0.30   0.07    40092.88  0.7639935  18684.92
##    0.30   0.08    40092.88  0.7639935  18684.92
##    0.35   0.00    39019.04  0.7745176  18731.17
##    0.35   0.01    39019.04  0.7745176  18731.17
##    0.35   0.02    39019.04  0.7745176  18731.17
##    0.35   0.03    39019.04  0.7745176  18731.17
##    0.35   0.04    39019.04  0.7745176  18731.17
##    0.35   0.05    39019.04  0.7745176  18731.17
##    0.35   0.06    39019.04  0.7745176  18731.17
##    0.35   0.07    39019.04  0.7745176  18731.17
##    0.35   0.08    39019.04  0.7745176  18731.17
##    0.40   0.00    38719.84  0.7772436  18699.61
##    0.40   0.01    38719.84  0.7772436  18699.61
##    0.40   0.02    38719.84  0.7772436  18699.61
##    0.40   0.03    38719.84  0.7772436  18699.61
##    0.40   0.04    38719.84  0.7772436  18699.61
##    0.40   0.05    38719.84  0.7772436  18699.61
##    0.40   0.06    38719.84  0.7772436  18699.61
##    0.40   0.07    38719.84  0.7772436  18699.61
##    0.40   0.08    38719.84  0.7772436  18699.61
##    0.45   0.00    38689.88  0.7776458  18711.54
##    0.45   0.01    38689.88  0.7776458  18711.54
##    0.45   0.02    38689.88  0.7776458  18711.54
##    0.45   0.03    38689.88  0.7776458  18711.54
##    0.45   0.04    38689.88  0.7776458  18711.54
##    0.45   0.05    38689.88  0.7776458  18711.54
##    0.45   0.06    38689.88  0.7776458  18711.54
##    0.45   0.07    38689.88  0.7776458  18711.54
##    0.45   0.08    38689.88  0.7776458  18711.54
##    0.50   0.00    38649.80  0.7781005  18710.15
##    0.50   0.01    38649.80  0.7781005  18710.15
##    0.50   0.02    38649.80  0.7781005  18710.15
##    0.50   0.03    38649.80  0.7781005  18710.15
##    0.50   0.04    38649.80  0.7781005  18710.15
##    0.50   0.05    38649.80  0.7781005  18710.15
##    0.50   0.06    38649.80  0.7781005  18710.15
##    0.50   0.07    38649.80  0.7781005  18710.15
##    0.50   0.08    38649.80  0.7781005  18710.15
##    0.55   0.00    38822.08  0.7767128  18748.51
##    0.55   0.01    38822.08  0.7767128  18748.51
##    0.55   0.02    38822.08  0.7767128  18748.51
##    0.55   0.03    38822.08  0.7767128  18748.51
##    0.55   0.04    38822.08  0.7767128  18748.51
##    0.55   0.05    38822.08  0.7767128  18748.51
##    0.55   0.06    38822.08  0.7767128  18748.51
##    0.55   0.07    38822.08  0.7767128  18748.51
##    0.55   0.08    38822.08  0.7767128  18748.51
##    0.60   0.00    38808.49  0.7768980  18750.57
##    0.60   0.01    38808.49  0.7768980  18750.57
##    0.60   0.02    38808.49  0.7768980  18750.57
```

```
## 0.60 0.03 38808.49 0.7768980 18750.57
## 0.60 0.04 38808.49 0.7768980 18750.57
## 0.60 0.05 38808.49 0.7768980 18750.57
## 0.60 0.06 38808.49 0.7768980 18750.57
## 0.60 0.07 38808.49 0.7768980 18750.57
## 0.60 0.08 38808.49 0.7768980 18750.57
## 0.65 0.00 38801.02 0.7770118 18751.30
## 0.65 0.01 38801.02 0.7770118 18751.30
## 0.65 0.02 38801.02 0.7770118 18751.30
## 0.65 0.03 38801.02 0.7770118 18751.30
## 0.65 0.04 38801.02 0.7770118 18751.30
## 0.65 0.05 38801.02 0.7770118 18751.30
## 0.65 0.06 38801.02 0.7770118 18751.30
## 0.65 0.07 38801.02 0.7770118 18751.30
## 0.65 0.08 38801.02 0.7770118 18751.30
## 0.70 0.00 38817.58 0.7769118 18767.54
## 0.70 0.01 38817.58 0.7769118 18767.54
## 0.70 0.02 38817.58 0.7769118 18767.54
## 0.70 0.03 38817.58 0.7769118 18767.54
## 0.70 0.04 38817.58 0.7769118 18767.54
## 0.70 0.05 38817.58 0.7769118 18767.54
## 0.70 0.06 38817.58 0.7769118 18767.54
## 0.70 0.07 38817.58 0.7769118 18767.54
## 0.70 0.08 38817.58 0.7769118 18767.54
## 0.75 0.00 38822.78 0.7768826 18750.66
## 0.75 0.01 38822.78 0.7768826 18750.66
## 0.75 0.02 38822.78 0.7768826 18750.66
## 0.75 0.03 38822.78 0.7768826 18750.66
## 0.75 0.04 38822.78 0.7768826 18750.66
## 0.75 0.05 38822.78 0.7768826 18750.66
## 0.75 0.06 38822.78 0.7768826 18750.66
## 0.75 0.07 38822.78 0.7768826 18750.66
## 0.75 0.08 38822.78 0.7768826 18750.66
## 0.80 0.00 38697.66 0.7779874 18718.54
## 0.80 0.01 38697.66 0.7779874 18718.54
## 0.80 0.02 38697.66 0.7779874 18718.54
## 0.80 0.03 38697.66 0.7779874 18718.54
## 0.80 0.04 38697.66 0.7779874 18718.54
## 0.80 0.05 38697.66 0.7779874 18718.54
## 0.80 0.06 38697.66 0.7779874 18718.54
## 0.80 0.07 38697.66 0.7779874 18718.54
## 0.80 0.08 38697.66 0.7779874 18718.54
## 0.85 0.00 38725.38 0.7777563 18711.19
## 0.85 0.01 38725.38 0.7777563 18711.19
## 0.85 0.02 38725.38 0.7777563 18711.19
## 0.85 0.03 38725.38 0.7777563 18711.19
## 0.85 0.04 38725.38 0.7777563 18711.19
## 0.85 0.05 38725.38 0.7777563 18711.19
## 0.85 0.06 38725.38 0.7777563 18711.19
## 0.85 0.07 38725.38 0.7777563 18711.19
## 0.85 0.08 38725.38 0.7777563 18711.19
## 0.90 0.00 38767.74 0.7773957 18708.69
## 0.90 0.01 38767.74 0.7773957 18708.69
## 0.90 0.02 38767.74 0.7773957 18708.69
```

```
##   0.90   0.03    38767.74  0.7773957  18708.69
##   0.90   0.04    38767.74  0.7773957  18708.69
##   0.90   0.05    38767.74  0.7773957  18708.69
##   0.90   0.06    38767.74  0.7773957  18708.69
##   0.90   0.07    38767.74  0.7773957  18708.69
##   0.90   0.08    38767.74  0.7773957  18708.69
##   0.95   0.00    38954.38  0.7757676  18739.75
##   0.95   0.01    38954.38  0.7757676  18739.75
##   0.95   0.02    38954.38  0.7757676  18739.75
##   0.95   0.03    38954.38  0.7757676  18739.75
##   0.95   0.04    38954.38  0.7757676  18739.75
##   0.95   0.05    38954.38  0.7757676  18739.75
##   0.95   0.06    38954.38  0.7757676  18739.75
##   0.95   0.07    38954.38  0.7757676  18739.75
##   0.95   0.08    38954.38  0.7757676  18739.75
##   1.00   0.00    38869.60  0.7765108  18707.08
##   1.00   0.01    38869.60  0.7765108  18707.08
##   1.00   0.02    38869.60  0.7765108  18707.08
##   1.00   0.03    38869.60  0.7765108  18707.08
##   1.00   0.04    38869.60  0.7765108  18707.08
##   1.00   0.05    38869.60  0.7765108  18707.08
##   1.00   0.06    38869.60  0.7765108  18707.08
##   1.00   0.07    38869.60  0.7765108  18707.08
##   1.00   0.08    38869.60  0.7765108  18707.08
##
## RMSE was used to select the optimal model using  the smallest value.
## The final values used for the model were alpha = 0 and lambda = 0.08.
```

```r
library(rminer)
```

```r
set.seed(100)
inTrain <- createDataPartition(train$SalePrice, p=0.7, list=FALSE)

str(inTrain)
```

```
##  int [1:1024, 1] 1 2 3 4 7 11 12 13 17 18 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr "Resample1"
```

```r
#inTrain

saleTrain <- train[inTrain,]
saleTest <- train[-inTrain,]

myTrainControl = trainControl(method = "cv", number = 5, verboseIter = FALSE)
fit.glmnet <- train(SalePrice~.,saleTrain,trControl = myTrainControl,
                    method="glmnet",tuneGrid=expand.grid(.alpha = seq(0,1,by=0.05),
                                                         .lambda = seq(0, 0.08, by = 0.01)))

predicted <- predict(fit.glmnet, saleTest)
```

```r
mmetric(saleTest$SalePrice, predicted, metric=c("RMSE","R2"))
```

```
##         RMSE           R2
## 2.912382e+04 8.663172e-01
```

```r
sqrt(mean((saleTest$SalePrice - predicted)^2))/mean(saleTest$SalePrice)
```

```
## [1] 0.1610607
```