Untitled

XiangluHe

12/1/2019

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------ tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
## Warning: package 'purrr' was built under R version 3.6.2
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
## -- Conflicts --------------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(sqldf)
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jre1.8.0_231')
library(rJava)
library(extraTrees)
library(ipred)

#Import Data Set and Data Cleaning
data_1 <- read_csv("bmw.csv")
```

```
## Parsed with column specification:
## cols(
##   maker_key = col_character(),
##   model_key = col_character(),
##   mileage = col_double(),
##   engine_power = col_double(),
##   registration_date = col_character(),
##   fuel = col_character(),
##   paint_color = col_character(),
##   car_type = col_character(),
##   feature_1 = col_logical(),
##   feature_2 = col_logical(),
##   feature_3 = col_logical(),
##   feature_4 = col_logical(),
##   feature_5 = col_logical(),
##   feature_6 = col_logical(),
##   feature_7 = col_logical(),
##   feature_8 = col_logical(),
##   price = col_double(),
##   sold_at = col_character(),
##   Model = col_character()
## )
```

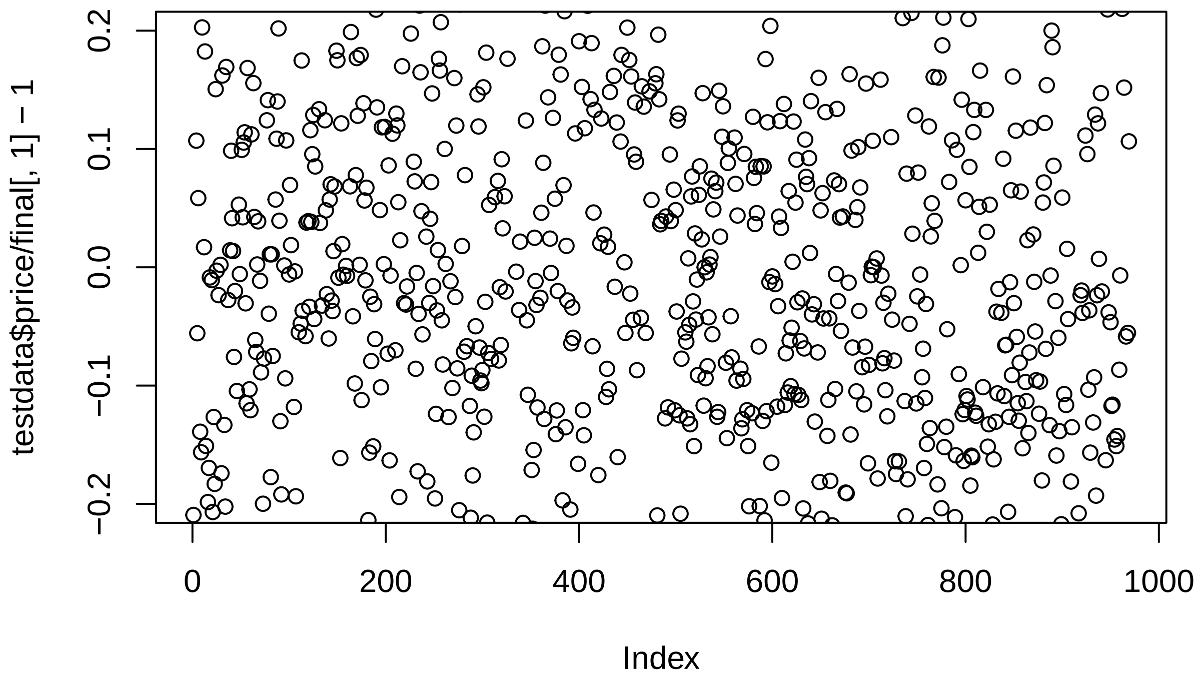
```
right = function(text, num_char) {
  substr(text, nchar(text) - (num_char-1), nchar(text))
}
data_1$registration_date <- right(data_1$registration_date,4)
data_1$registration_date <- as.numeric(data_1$registration_date)


#Spilt Train and Test Set
BMW <- data_1
BMW$ID <- c(1:nrow(BMW))
set.seed(50)
training <- BMW[sample(1:nrow(BMW), round(0.8*nrow(BMW),0), replace=FALSE),]
testdata <- sqldf("select * from BMW where ID not in (select ID from training)")
testdata <- data.frame(testdata)



#Factoring
training$model_key <- as.factor(training$model_key)
training$fuel <- as.factor(training$fuel)
training$car_type <- as.factor(training$car_type)
training$feature_1 <- as.numeric(training$feature_1)
training$feature_2 <- as.numeric(training$feature_2)
training$feature_3 <- as.numeric(training$feature_3)
training$feature_4 <- as.numeric(training$feature_4)
training$feature_5 <- as.numeric(training$feature_5)
training$feature_6 <- as.numeric(training$feature_6)
training$feature_7 <- as.numeric(training$feature_7)
training$feature_8 <- as.numeric(training$feature_8)
training$Model <- as.factor(training$Model)

#Create dependents and independt variables
y <- training$price
x1 <- training$model_key
x2 <- training$mileage
x3 <- training$engine_power
x4 <- training$registration_date
x5 <- training$car_type
x6 <- training$feature_1
```

```r
x7 <- training$feature_2
x8 <- training$feature_3
x9 <- training$feature_4
x10 <- training$feature_5
x11 <- training$feature_6
x12 <- training$feature_7
x13 <- training$feature_8
#x14 <- training$Model
x15 <- training$fuel
x16 <- training$paint_color


#Run multiple regression model One
model <- lm(y~x2*x4+x3+x5+x6+x7+x8+x9+x10+x11+x12+x13+x15)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x2 * x4 + x3 + x5 + x6 + x7 + x8 + x9 + x10 +
##     x11 + x12 + x13 + x15)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -26613  -2007      0   1942 157627
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -4.518e+06  1.690e+05 -26.731  < 2e-16 ***
## x2                1.326e+01  7.560e-01  17.535  < 2e-16 ***
## x4                2.248e+03  8.398e+01  26.768  < 2e-16 ***
## x3                1.050e+02  2.970e+00  35.358  < 2e-16 ***
## x5coupe          -9.317e+02  1.025e+03  -0.909 0.363420
## x5estate         -4.309e+03  8.885e+02  -4.850 1.28e-06 ***
## x5hatchback      -3.166e+03  9.034e+02  -3.504 0.000463 ***
## x5sedan          -1.862e+03  8.876e+02  -2.098 0.035994 *
## x5subcompact     -2.852e+03  1.039e+03  -2.744 0.006090 **
## x5suv            -1.232e+02  9.049e+02  -0.136 0.891714
## x5van            -6.580e+03  1.244e+03  -5.288 1.31e-07 ***
## x6                1.355e+03  1.968e+02   6.883 6.82e-12 ***
## x7                1.011e+03  2.559e+02   3.951 7.93e-05 ***
## x8                1.069e+03  2.288e+02   4.672 3.08e-06 ***
## x9                9.674e+02  2.758e+02   3.508 0.000457 ***
## x10               6.054e+01  1.990e+02   0.304 0.760971
## x11               6.365e+02  2.130e+02   2.988 0.002828 **
## x12               8.454e+02  3.931e+02   2.150 0.031581 *
## x13               1.646e+03  2.068e+02   7.960 2.25e-15 ***
## x15electro        3.698e+03  3.061e+03   1.208 0.227071
## x15hybrid_petrol  1.258e+04  1.876e+03   6.708 2.26e-11 ***
## x15petrol        -1.154e+03  4.639e+02  -2.488 0.012889 *
## x2:x4            -6.607e-03  3.758e-04 -17.580  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 5275 on 3851 degrees of freedom
## Multiple R-squared:  0.6855, Adjusted R-squared:  0.6837
## F-statistic: 381.6 on 22 and 3851 DF,  p-value: < 2.2e-16
```

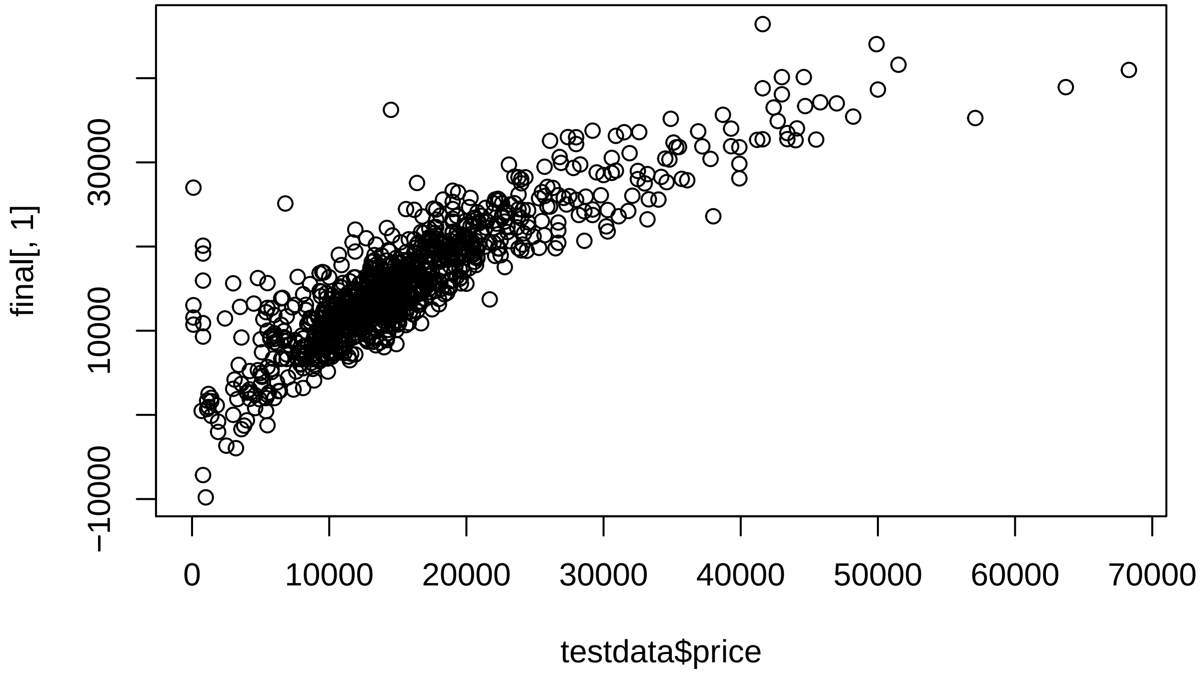
```
#Test the model
test1<- testdata
test12 <- data.frame(x2=test1$mileage,x3=test1$engine_power,x4=test1$registration_date,x5=test1$car_typ
test12$x6 <- ifelse(test12$x6=='TRUE',1,0)
test12$x7 <- ifelse(test12$x7=='TRUE',1,0)
test12$x8 <- ifelse(test12$x8=='TRUE',1,0)
test12$x9 <- ifelse(test12$x9=='TRUE',1,0)
test12$x10 <- ifelse(test12$x10=='TRUE',1,0)
test12$x11 <- ifelse(test12$x11=='TRUE',1,0)
test12$x12 <- ifelse(test12$x12=='TRUE',1,0)
test12$x13 <- ifelse(test12$x13=='TRUE',1,0)
test12<- tbl_df(test12)
final <- predict(model,test12,interval = 'prediction',level=0.99)
final <- final[,1]
final <- data.frame(final)


#Plot the percent difference between predict and actual values in test set
plot(testdata$price/final[,1]-1,ylim=c(-0.2,0.2))
```



```r
result <- (testdata$price/final)-1
result <- data.frame(result)
good <- result %>%
  filter(result <= 0.2 & result >-0.2)

#Percent within 20 difference
nrow(good)/nrow(testdata)
```

```
## [1] 0.6429309
```

```r
plot(testdata$price,final[,1])
```



```r
sum(model$fitted.values/training$price-1<0.1)/length(training$price)
```

```
## [1] 0.6843056
```

```r
#Predit Training test
finaltri <- predict(model,data.frame(x2 <- training$mileage
,x4 <- training$registration_date
,x5 <- training$car_type
,x6 <- training$feature_1
,x7 <- training$feature_2
,x8 <- training$feature_3
,x9 <- training$feature_4
,x10 <- training$feature_5
,x11 <- training$feature_6
,x12 <- training$feature_7
,x13 <- training$feature_8
,x15 <- training$fuel))
#finaltri <- finaltri[,1]
finaltri <- data.frame(finaltri)

result <- (training$price/finaltri)-1
result <- data.frame(result)
good <- result %>%
  filter(result <= 0.2 & result >-0.2)

#Percent within 20 difference
nrow(good)/nrow(training)
```

```
## [1] 0.6621064
```

##MLR Model Two

```
#Import Data Set and Data Cleaning
data_1 <- read_csv("bmw.csv")
```

```
## Parsed with column specification:
## cols(
##   maker_key = col_character(),
##   model_key = col_character(),
##   mileage = col_double(),
##   engine_power = col_double(),
##   registration_date = col_character(),
##   fuel = col_character(),
##   paint_color = col_character(),
##   car_type = col_character(),
##   feature_1 = col_logical(),
##   feature_2 = col_logical(),
##   feature_3 = col_logical(),
##   feature_4 = col_logical(),
##   feature_5 = col_logical(),
##   feature_6 = col_logical(),
##   feature_7 = col_logical(),
##   feature_8 = col_logical(),
##   price = col_double(),
##   sold_at = col_character(),
##   Model = col_character()
## )
```

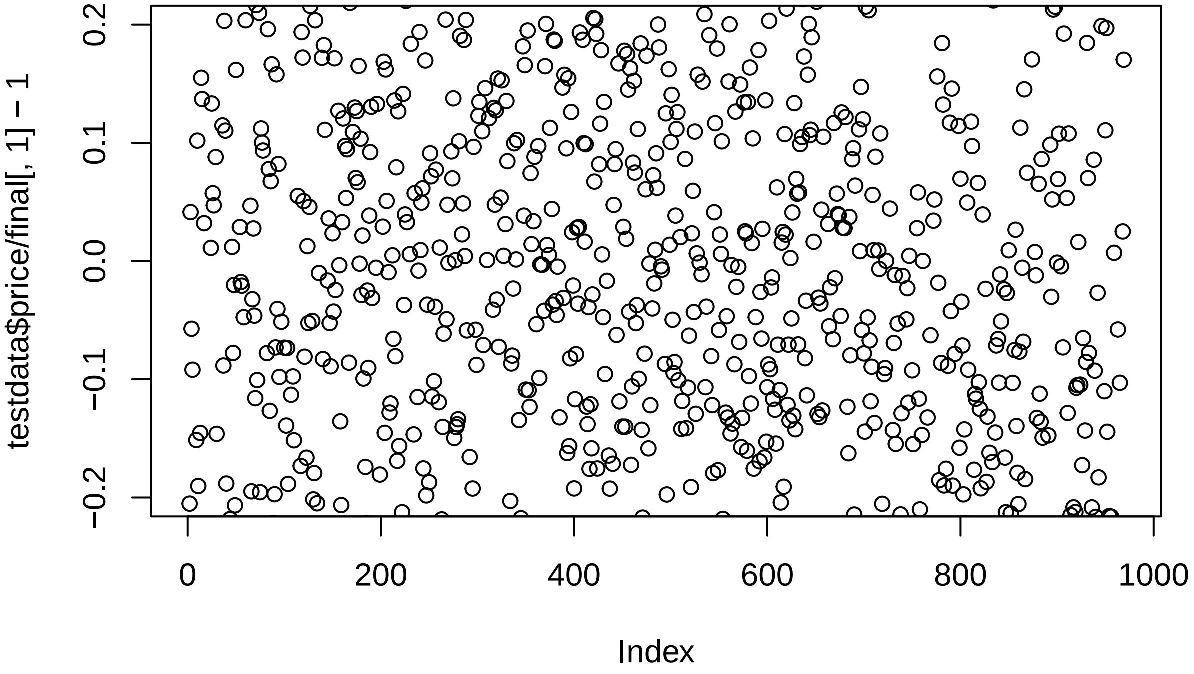
```
right = function(text, num_char) {
  substr(text, nchar(text) - (num_char-1), nchar(text))
}
data_1$registration_date <- right(data_1$registration_date,4)
data_1$registration_date <- as.numeric(data_1$registration_date)
```

```
BMW <- data_1
BMW$ID <- c(1:nrow(BMW))
set.seed(35)
training <- BMW[sample(1:nrow(BMW), round(0.8*nrow(BMW),0), replace=FALSE),]
testdata <- sqldf("select * from BMW where ID not in (select ID from training)")
testdata <- data.frame(testdata)



#Factoring
training$model_key <- as.factor(training$model_key)
training$fuel <- as.factor(training$fuel)
training$car_type <- as.factor(training$car_type)
training$feature_1 <- as.numeric(training$feature_1)
training$feature_2 <- as.numeric(training$feature_2)
training$feature_3 <- as.numeric(training$feature_3)
training$feature_4 <- as.numeric(training$feature_4)
training$feature_5 <- as.numeric(training$feature_5)
training$feature_6 <- as.numeric(training$feature_6)
training$feature_7 <- as.numeric(training$feature_7)
training$feature_8 <- as.numeric(training$feature_8)
training$Model <- as.factor(training$Model)

#Create dependents and independt variables
y <- training$price
x1 <- training$model_key
x2 <- training$mileage
x3 <- training$engine_power
x4 <- training$registration_date
x5 <- training$car_type
x6 <- training$feature_1
x7 <- training$feature_2
x8 <- training$feature_3
x9 <- training$feature_4
x10 <- training$feature_5
x11 <- training$feature_6
x12 <- training$feature_7
x13 <- training$feature_8
#x14 <- training$Model
x15 <- training$fuel
x16 <- training$paint_color
#Run multiple regression model Two
model <- lm(y~x2+x3+x4+x5+x6+x8+x9+x10+x11+x13+x15)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x2 + x3 + x4 + x5 + x6 + x8 + x9 + x10 + x11 +
##     x13 + x15)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -27398  -2080   -133   1804 120375
```

```
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.079e+06  7.855e+04 -26.465  < 2e-16 ***
## x2                -3.197e-02  1.587e-03 -20.136  < 2e-16 ***
## x3                 1.020e+02  2.736e+00  37.275  < 2e-16 ***
## x4                 1.036e+03  3.902e+01  26.558  < 2e-16 ***
## x5coupe           -2.734e+02  9.900e+02  -0.276 0.782431
## x5estate          -3.982e+03  8.480e+02  -4.696 2.75e-06 ***
## x5hatchback       -2.572e+03  8.624e+02  -2.983 0.002874 **
## x5sedan           -1.566e+03  8.465e+02  -1.850 0.064374 .
## x5subcompact      -2.145e+03  9.821e+02  -2.184 0.028996 *
## x5suv             -3.435e+02  8.621e+02  -0.398 0.690304
## x5van             -4.830e+03  1.174e+03  -4.115 3.96e-05 ***
## x6                 1.709e+03  1.757e+02   9.730  < 2e-16 ***
## x8                 1.025e+03  2.110e+02   4.859 1.23e-06 ***
## x9                 1.732e+03  2.528e+02   6.851 8.52e-12 ***
## x10               -3.840e+01  1.798e+02  -0.214 0.830926
## x11                7.536e+02  1.950e+02   3.864 0.000113 ***
## x13                1.740e+03  1.883e+02   9.240  < 2e-16 ***
## x15electro         6.553e+03  4.890e+03   1.340 0.180284
## x15hybrid_petrol   1.302e+04  1.854e+03   7.023 2.56e-12 ***
## x15petrol         -1.466e+03  4.279e+02  -3.425 0.000621 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4880 on 3854 degrees of freedom
## Multiple R-squared:  0.7001, Adjusted R-squared:  0.6986
## F-statistic: 473.6 on 19 and 3854 DF,  p-value: < 2.2e-16
```

```
#Test the model
test1<- testdata
test12 <- data.frame(x2=test1$mileage,x3=test1$engine_power,x4=test1$registration_date,x5=test1$car_type
test12$x6 <- ifelse(test12$x6=='TRUE',1,0)
#test12$x7 <- ifelse(test12$x7=='TRUE',1,0)
test12$x8 <- ifelse(test12$x8=='TRUE',1,0)
test12$x9 <- ifelse(test12$x9=='TRUE',1,0)
test12$x10 <- ifelse(test12$x10=='TRUE',1,0)
test12$x11 <- ifelse(test12$x11=='TRUE',1,0)
#test12$x12 <- ifelse(test12$x12=='TRUE',1,0)
test12$x13 <- ifelse(test12$x13=='TRUE',1,0)
test12<- tbl_df(test12)
final <- predict(model,test12,interval = 'prediction',level=0.99)
final <- final[,1]
final <- data.frame(final)


#Plot the percent difference between predict and actual values in test set
plot(testdata$price/final[,1]-1,ylim=c(-0.2,0.2))
```



```
result <- (testdata$price/final)-1
result <- data.frame(result)
good <- result %>%
  filter(result <= 0.2 & result >-0.2)

#Percent within 20 difference
nrow(good)/nrow(testdata)
```

```
## [1] 0.6418989
```

```
finaltri <- predict(model,data.frame(x2=x2,x3=x3,x4=x4,x5=x5,x6=x6,x8=x8,x9=x9,x10=x10,x11=x12,x13=x13,
finaltri <- finaltri[,1]
finaltri <- data.frame(finaltri)

result <- (training$price/finaltri)-1
result <- data.frame(result)
good <- result %>%
  filter(result <= 0.2 & result >-0.2)

#Percent within 20 difference in training set
nrow(good)/nrow(training)
```

```
## [1] 0.683015
```

SVM Model One

```
library(e1071)
svmdata <- read_csv('bmw.csv')
```

```
## Parsed with column specification:
## cols(
##   maker_key = col_character(),
##   model_key = col_character(),
##   mileage = col_double(),
##   engine_power = col_double(),
##   registration_date = col_character(),
##   fuel = col_character(),
##   paint_color = col_character(),
##   car_type = col_character(),
##   feature_1 = col_logical(),
##   feature_2 = col_logical(),
##   feature_3 = col_logical(),
##   feature_4 = col_logical(),
##   feature_5 = col_logical(),
##   feature_6 = col_logical(),
##   feature_7 = col_logical(),
##   feature_8 = col_logical(),
##   price = col_double(),
##   sold_at = col_character(),
##   Model = col_character()
## )
```

```
right = function(text, num_char) {
  substr(text, nchar(text) - (num_char-1), nchar(text))
}
svmdata$registration_date <- right(svmdata$registration_date,4)
svmdata$registration_date <- as.numeric(svmdata$registration_date)
svmdata <- svmdata %>%
  filter(svmdata$engine_power > 0)

#Spilt Train and Test Set
svmdata$ID <- c(1:nrow(svmdata))
set.seed(60)
trainingsvm <- svmdata[sample(1:nrow(svmdata), round(0.8*nrow(svmdata),0), replace=FALSE),]
testdatasvm <- sqldf("select * from svmdata where ID not in (select ID from trainingsvm)")
testdatasvm <- data.frame(testdatasvm)

price <- trainingsvm$price
mileage <-trainingsvm$mileage
engine_power <- trainingsvm$engine_power
date <- trainingsvm$registration_date
type <- trainingsvm$car_type
fuel <- trainingsvm$fuel
f1 <-trainingsvm$feature_1
f2 <- trainingsvm$feature_2
f3 <- trainingsvm$feature_3
```

```r
f4 <- trainingsvm$feature_4
f5 <- trainingsvm$feature_5
f6 <-trainingsvm$feature_6
f7 <- trainingsvm$feature_7
f8 <- trainingsvm$feature_8

svmmodel <- svm(price~type+mileage*date+engine_power+f1+f2+f3+f4+f5+f6+f7+f8)
summary(svmmodel)
```

```
##
## Call:
## svm(formula = price ~ type + mileage * date + engine_power +
##     f1 + f2 + f3 + f4 + f5 + f6 + f7 + f8)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.05
##     epsilon:  0.1
##
##
## Number of Support Vectors:  2500
```

```r
svmpre1 <- predict(svmmodel,data.frame(type=testdatasvm$car_type,mileage=testdatasvm$mileage,engine_powe
svmpre1 <- data.frame(svmpre1)
result <- (testdatasvm$price/svmpre1$svmpre1)-1
result <- data.frame(result)
good <- result %>%
  filter(result <= 0.2 & result >-0.2)
nrow(good)/nrow(result)
```

```
## [1] 0.7716942
```

```r
svmpretri <- predict(svmmodel,data.frame(type=trainingsvm$car_type,mileage=trainingsvm$mileage,engine_po
svmpretri <- data.frame(svmpretri)
result <- (trainingsvm$price/svmpretri$svmpretri)-1
result <- data.frame(result)
good <- result %>%
  filter(result <= 0.2 & result >-0.2)


#Percent within 20 difference in training set
nrow(good)/nrow(trainingsvm)
```

```
## [1] 0.7934951
```

SVM Model Two

```r
price <- trainingsvm$price
mileage <-trainingsvm$mileage
engine_power <- trainingsvm$engine_power
date <- trainingsvm$registration_date
type <- trainingsvm$car_type
fuel <- trainingsvm$fuel
f1 <-trainingsvm$feature_1
f2 <- trainingsvm$feature_2
f3 <- trainingsvm$feature_3
f4 <- trainingsvm$feature_4
f5 <- trainingsvm$feature_5
f6 <-trainingsvm$feature_6
f7 <- trainingsvm$feature_7
f8 <- trainingsvm$feature_8
fuel <- ifelse(trainingsvm$fuel=='diesel',1,0)

svmmodel <- svm(price~type+mileage*date+engine_power+f1+f2+f3+f4+f5+f6+f7+f8+fuel)
summary(svmmodel)
```

```
##
## Call:
## svm(formula = price ~ type + mileage * date + engine_power +
##     f1 + f2 + f3 + f4 + f5 + f6 + f7 + f8 + fuel)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.04761905
##     epsilon:  0.1
##
##
## Number of Support Vectors:  2489
```

```r
testdatasvm$fuel <- ifelse(testdatasvm$fuel=='diesel',1,0)

svmpre1 <- predict(svmmodel,data.frame(type=testdatasvm$car_type,mileage=testdatasvm$mileage,date=testda
svmpre1 <- data.frame(svmpre1)
result <- (testdatasvm$price/svmpre1$svmpre1)-1
result <- data.frame(result)
good <- result %>%
  filter(result <= 0.2 & result >-0.2)
nrow(good)/nrow(result)
```

```
## [1] 0.7727273
```