Predict Salse of the Computer

#loading data  
library(readr)  
data<-read.csv("E:\\assignments data\\Computer\_Data (1).csv")  
#dataset details and EDA  
colnames(data)

## [1] "X" "price" "speed" "hd" "ram" "screen" "cd"   
## [8] "multi" "premium" "ads" "trend"

computerData<-data[, c(2:11)]  
attach(computerData)  
dim(computerData)

## [1] 6259 10

sum(is.null(computerData))

## [1] 0

library(plyr)  
computerData$cd<-revalue(computerData$cd, c("yes"="1", "no"="2"))  
computerData$multi<-revalue(computerData$multi, c("yes"="1", "no"="2"))  
computerData$premium<-revalue(computerData$premium, c("yes", "no"="2"))

## The following `from` values were not present in `x`:

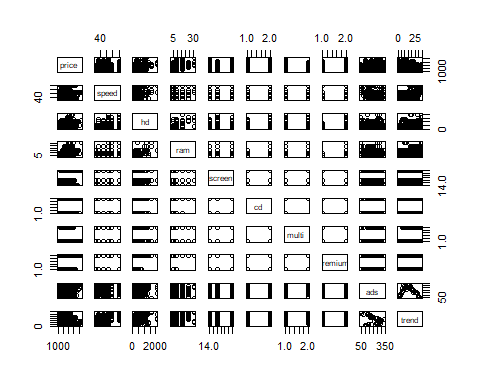
summary(computerData)

## price speed hd ram   
## Min. : 949 Min. : 25.00 Min. : 80.0 Min. : 2.000   
## 1st Qu.:1794 1st Qu.: 33.00 1st Qu.: 214.0 1st Qu.: 4.000   
## Median :2144 Median : 50.00 Median : 340.0 Median : 8.000   
## Mean :2220 Mean : 52.01 Mean : 416.6 Mean : 8.287   
## 3rd Qu.:2595 3rd Qu.: 66.00 3rd Qu.: 528.0 3rd Qu.: 8.000   
## Max. :5399 Max. :100.00 Max. :2100.0 Max. :32.000   
## screen cd multi premium ads trend   
## Min. :14.00 2:3351 2:5386 2 : 612 Min. : 39.0 Min. : 1.00   
## 1st Qu.:14.00 1:2908 1: 873 yes:5647 1st Qu.:162.5 1st Qu.:10.00   
## Median :14.00 Median :246.0 Median :16.00   
## Mean :14.61 Mean :221.3 Mean :15.93   
## 3rd Qu.:15.00 3rd Qu.:275.0 3rd Qu.:21.50   
## Max. :17.00 Max. :339.0 Max. :35.00

head(computerData)

## price speed hd ram screen cd multi premium ads trend  
## 1 1499 25 80 4 14 2 2 yes 94 1  
## 2 1795 33 85 2 14 2 2 yes 94 1  
## 3 1595 25 170 4 15 2 2 yes 94 1  
## 4 1849 25 170 8 14 2 2 2 94 1  
## 5 3295 33 340 16 14 2 2 yes 94 1  
## 6 3695 66 340 16 14 2 2 yes 94 1

# checking for multicollinearity  
plot(computerData)



computerData$cd<-as.numeric(computerData$cd)  
computerData$multi<-as.numeric(computerData$multi)  
computerData$premium<-as.numeric(computerData$premium)  
cor(computerData)

## price speed hd ram screen  
## price 1.00000000 0.30097646 0.43025779 0.62274824 0.296041474  
## speed 0.30097646 1.00000000 0.37230410 0.23476050 0.189074122  
## hd 0.43025779 0.37230410 1.00000000 0.77772630 0.232801530  
## ram 0.62274824 0.23476050 0.77772630 1.00000000 0.208953740  
## screen 0.29604147 0.18907412 0.23280153 0.20895374 1.000000000  
## cd 0.19734334 0.25825980 0.50357041 0.43850441 0.129487662  
## multi -0.01665139 0.08417193 0.09280483 0.04549689 -0.001740414  
## premium -0.08069636 0.11420791 0.19692359 0.19714459 0.018745223  
## ads 0.05454047 -0.21523206 -0.32322200 -0.18166971 -0.093919429  
## trend -0.19998694 0.40543833 0.57779013 0.27684384 0.188614445  
## cd multi premium ads trend  
## price 0.19734334 -0.016651388 -0.08069636 0.05454047 -0.19998694  
## speed 0.25825980 0.084171934 0.11420791 -0.21523206 0.40543833  
## hd 0.50357041 0.092804830 0.19692359 -0.32322200 0.57779013  
## ram 0.43850441 0.045496894 0.19714459 -0.18166971 0.27684384  
## screen 0.12948766 -0.001740414 0.01874522 -0.09391943 0.18861444  
## cd 1.00000000 0.432179298 0.21607660 -0.06109108 0.44578018  
## multi 0.43217930 1.000000000 0.12477474 -0.03039426 0.21090743  
## premium 0.21607660 0.124774741 1.00000000 -0.15202274 0.04210738  
## ads -0.06109108 -0.030394260 -0.15202274 1.00000000 -0.31855251  
## trend 0.44578018 0.210907431 0.04210738 -0.31855251 1.00000000

# pairwise correlation between predictor variables are low  
# so there is no multicollinearity  
  
library(corpcor)  
cor2pcor(cor(computerData))

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 1.0000000 0.537326713 0.33716508 0.49690857 0.36279673 0.080719005  
## [2,] 0.5373267 1.000000000 -0.10433213 -0.28438702 -0.11491585 -0.008311233  
## [3,] 0.3371651 -0.104332126 1.00000000 0.43156783 -0.10045528 0.087223167  
## [4,] 0.4969086 -0.284387018 0.43156783 1.00000000 -0.11960849 0.104933080  
## [5,] 0.3627967 -0.114915854 -0.10045528 -0.11960849 1.00000000 -0.024623811  
## [6,] 0.0807190 -0.008311233 0.08722317 0.10493308 -0.02462381 1.000000000  
## [7,] 0.1148699 -0.073810359 -0.12420636 -0.10279078 -0.07426269 0.400039320  
## [8,] -0.4626969 0.292808267 0.20637502 0.25151338 0.14359050 0.155633504  
## [9,] 0.1599485 -0.140381854 -0.20186341 -0.04815123 -0.06683674 0.164413496  
## [10,] -0.7219155 0.504953018 0.55265421 0.15896659 0.30166432 0.214237661  
## [,7] [,8] [,9] [,10]  
## [1,] 0.11486991 -0.46269692 0.159948478 -0.721915473  
## [2,] -0.07381036 0.29280827 -0.140381854 0.504953018  
## [3,] -0.12420636 0.20637502 -0.201863412 0.552654205  
## [4,] -0.10279078 0.25151338 -0.048151226 0.158966590  
## [5,] -0.07426269 0.14359050 -0.066836740 0.301664320  
## [6,] 0.40003932 0.15563350 0.164413496 0.214237661  
## [7,] 1.00000000 0.10912193 -0.049839360 0.141574261  
## [8,] 0.10912193 1.00000000 -0.046911259 -0.423282587  
## [9,] -0.04983936 -0.04691126 1.000000000 -0.001856949  
## [10,] 0.14157426 -0.42328259 -0.001856949 1.000000000

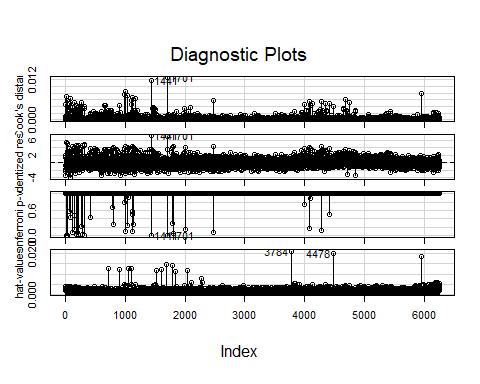
# finding influence factors  
model<-lm(price ~., data=computerData)  
summary(model)

##   
## Call:  
## lm(formula = price ~ ., data = computerData)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1093.77 -174.24 -11.49 146.49 2001.05   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 651.97219 64.49224 10.109 < 2e-16 \*\*\*  
## speed 9.32028 0.18506 50.364 < 2e-16 \*\*\*  
## hd 0.78178 0.02761 28.311 < 2e-16 \*\*\*  
## ram 48.25596 1.06608 45.265 < 2e-16 \*\*\*  
## screen 123.08904 3.99950 30.776 < 2e-16 \*\*\*  
## cd 60.91671 9.51559 6.402 1.65e-10 \*\*\*  
## multi 104.32382 11.41268 9.141 < 2e-16 \*\*\*  
## premium -509.22473 12.34225 -41.259 < 2e-16 \*\*\*  
## ads 0.65729 0.05132 12.809 < 2e-16 \*\*\*  
## trend -51.84958 0.62871 -82.470 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 275.3 on 6249 degrees of freedom  
## Multiple R-squared: 0.7756, Adjusted R-squared: 0.7752   
## F-statistic: 2399 on 9 and 6249 DF, p-value: < 2.2e-16

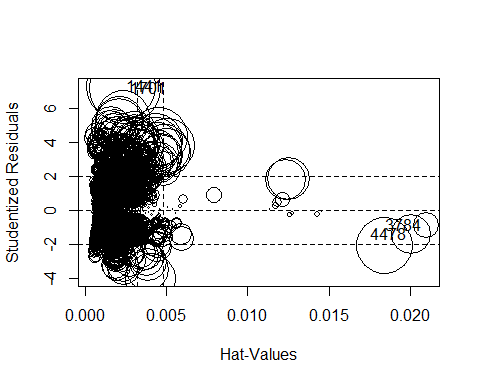
library(car)

## Loading required package: carData

influenceIndexPlot(model)



influencePlot(model)



## StudRes Hat CookD  
## 1441 7.3058529 0.002228075 0.011819949  
## 1701 7.1838002 0.002464463 0.012647347  
## 3784 -0.8667018 0.020972880 0.001609237  
## 4478 -1.3795547 0.020060286 0.003895407

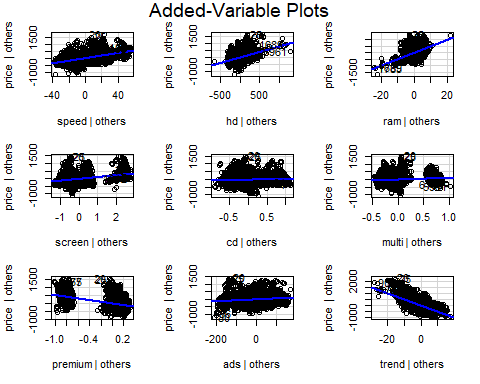
model1<-lm(price ~., data=computerData[-c(1441,1701,3784,4478),])  
summary(model1)

##   
## Call:  
## lm(formula = price ~ ., data = computerData[-c(1441, 1701, 3784,   
## 4478), ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1094.21 -173.10 -10.94 146.35 1509.23   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 681.81418 64.02437 10.649 < 2e-16 \*\*\*  
## speed 9.29551 0.18355 50.642 < 2e-16 \*\*\*  
## hd 0.78355 0.02795 28.030 < 2e-16 \*\*\*  
## ram 48.29482 1.06764 45.235 < 2e-16 \*\*\*  
## screen 121.07383 3.97118 30.488 < 2e-16 \*\*\*  
## cd 60.31315 9.44029 6.389 1.79e-10 \*\*\*  
## multi 104.85186 11.31912 9.263 < 2e-16 \*\*\*  
## premium -510.03064 12.24104 -41.666 < 2e-16 \*\*\*  
## ads 0.65465 0.05095 12.849 < 2e-16 \*\*\*  
## trend -51.74355 0.62676 -82.558 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 273.1 on 6245 degrees of freedom  
## Multiple R-squared: 0.7774, Adjusted R-squared: 0.777   
## F-statistic: 2423 on 9 and 6245 DF, p-value: < 2.2e-16

vif(model1)

## speed hd ram screen cd multi premium ads   
## 1.265374 4.324143 3.031458 1.081753 1.859410 1.290615 1.109459 1.219746   
## trend   
## 2.043377

# vif<10 so there is no multicollinearity  
avPlots(model1)



# fitting of multiple linear regression model  
model2<-lm(price ~ .-computerData$cd-computerData$multi, data=computerData[-c(1441,1701,3784,4478)])  
summary(model2)

##   
## Call:  
## lm(formula = price ~ . - computerData$cd - computerData$multi,   
## data = computerData[-c(1441, 1701, 3784, 4478)])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1093.77 -174.24 -11.49 146.49 2001.05   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 651.97219 64.49224 10.109 < 2e-16 \*\*\*  
## speed 9.32028 0.18506 50.364 < 2e-16 \*\*\*  
## hd 0.78178 0.02761 28.311 < 2e-16 \*\*\*  
## ram 48.25596 1.06608 45.265 < 2e-16 \*\*\*  
## screen 123.08904 3.99950 30.776 < 2e-16 \*\*\*  
## cd 60.91671 9.51559 6.402 1.65e-10 \*\*\*  
## multi 104.32382 11.41268 9.141 < 2e-16 \*\*\*  
## premium -509.22473 12.34225 -41.259 < 2e-16 \*\*\*  
## ads 0.65729 0.05132 12.809 < 2e-16 \*\*\*  
## trend -51.84958 0.62871 -82.470 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 275.3 on 6249 degrees of freedom  
## Multiple R-squared: 0.7756, Adjusted R-squared: 0.7752   
## F-statistic: 2399 on 9 and 6249 DF, p-value: < 2.2e-16

predicted\_values<-predict(model2)  
head(predicted\_values)

## 1 2 3 4 5 6   
## 2020.519 2002.478 2213.968 2793.128 2877.415 3184.985

# here p values <0.05  
#multiple R-squared value is 0.7756  
#Adjusted R-squared value is 0.7752  
par(mfrow=c(2,2))  
plot(model2)

