Computer Price Prediction using Stepwise Linear Regression & Feature Selection

libraries <- c(‘MASS’,‘leaps’, ‘FNN’) # install.packages(libraries)

# Loading dataframe

c\_prices <- read.csv("C:/Users/Arup/Documents/DS\_ComputerConfigure.csv",stringsAsFactors = FALSE)   
str(c\_prices)

## 'data.frame': 6259 obs. of 11 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ price : int 1499 1795 1595 1849 3295 3695 1720 1995 2225 2575 ...  
## $ speed : int 25 33 25 25 33 66 25 50 50 50 ...  
## $ hd : int 80 85 170 170 340 340 170 85 210 210 ...  
## $ ram : int 4 2 4 8 16 16 4 2 8 4 ...  
## $ screen : int 14 14 15 14 14 14 14 14 14 15 ...  
## $ cd : chr "no" "no" "no" "no" ...  
## $ multi : chr "no" "no" "no" "no" ...  
## $ premium: chr "yes" "yes" "yes" "no" ...  
## $ ads : int 94 94 94 94 94 94 94 94 94 94 ...  
## $ trend : int 1 1 1 1 1 1 1 1 1 1 ...

# Numerizing the Dataset

c\_prices<-c\_prices[-1]  
  
c\_prices[which(c\_prices$cd == "no"),]$cd <- 0; c\_prices[which(c\_prices$cd == "yes"),]$cd <- 1  
c\_prices[which(c\_prices$multi == "no"),]$multi <- 0; c\_prices[which(c\_prices$multi == "yes"),]$multi <- 1  
c\_prices[which(c\_prices$premium == "no"),]$premium <- 0; c\_prices[which(c\_prices$premium == "yes"),]$premium <- 1  
  
  
c\_prices$cd<-as.integer(c\_prices$cd)  
c\_prices$multi<-as.integer(c\_prices$multi)  
c\_prices$premium<-as.integer(c\_prices$premium)  
  
head(c\_prices)

## price speed hd ram screen cd multi premium ads trend  
## 1 1499 25 80 4 14 0 0 1 94 1  
## 2 1795 33 85 2 14 0 0 1 94 1  
## 3 1595 25 170 4 15 0 0 1 94 1  
## 4 1849 25 170 8 14 0 0 0 94 1  
## 5 3295 33 340 16 14 0 0 1 94 1  
## 6 3695 66 340 16 14 0 0 1 94 1

# Feature (Attribute) Selection using Forward Stepwise Regression

library(MASS) # stepwise regression

## Warning: package 'MASS' was built under R version 4.1.3

full <- lm(price ~ speed + hd + ram + screen + cd + multi + premium + ads + trend, data=c\_prices)  
null <- lm(price~1,data=c\_prices)  
  
stepF <- stepAIC(null, scope=list(lower=null, upper=full), direction= "forward", trace=TRUE)

## Start: AIC=79670.73  
## price ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + ram 1 818690431 1292340953 76601  
## + hd 1 390797865 1720233519 78391  
## + speed 1 191231639 1919799745 79078  
## + screen 1 185011960 1926019424 79099  
## + trend 1 84430224 2026601160 79417  
## + cd 1 82212838 2028818546 79424  
## + premium 1 13746829 2097284555 79632  
## + ads 1 6279607 2104751777 79654  
## <none> 2111031384 79671  
## + multi 1 585323 2110446061 79671  
##   
## Step: AIC=76601.3  
## price ~ ram  
##   
## Df Sum of Sq RSS AIC  
## + trend 1 317046568 975294385 74842  
## + premium 1 90928941 1201412013 76147  
## + ads 1 61377109 1230963845 76299  
## + screen 1 60765788 1231575165 76302  
## + speed 1 53523313 1238817640 76339  
## + hd 1 15618983 1276721971 76527  
## + cd 1 14990800 1277350154 76530  
## + multi 1 4280755 1288060198 76583  
## <none> 1292340953 76601  
##   
## Step: AIC=74841.57  
## price ~ ram + trend  
##   
## Df Sum of Sq RSS AIC  
## + speed 1 219800193 755494193 73245  
## + screen 1 107619693 867674692 74112  
## + premium 1 95496579 879797806 74199  
## + hd 1 70850709 904443676 74372  
## + cd 1 9234744 966059642 74784  
## + ads 1 8402231 966892154 74789  
## + multi 1 2706117 972588268 74826  
## <none> 975294385 74842  
##   
## Step: AIC=73245.23  
## price ~ ram + trend + speed  
##   
## Df Sum of Sq RSS AIC  
## + premium 1 121458788 634035405 72150  
## + screen 1 78678269 676815924 72559  
## + hd 1 44107493 711386700 72871  
## + ads 1 17590316 737903876 73100  
## + cd 1 5471868 750022325 73202  
## + multi 1 2685619 752808574 73225  
## <none> 755494193 73245  
##   
## Step: AIC=72150.23  
## price ~ ram + trend + speed + premium  
##   
## Df Sum of Sq RSS AIC  
## + screen 1 72874490 561160915 71388  
## + hd 1 58613852 575421553 71545  
## + cd 1 17368943 616666462 71978  
## + multi 1 9176056 624859350 72061  
## + ads 1 8152702 625882703 72071  
## <none> 634035405 72150  
##   
## Step: AIC=71388.02  
## price ~ ram + trend + speed + premium + screen  
##   
## Df Sum of Sq RSS AIC  
## + hd 1 54901344 506259571 70746  
## + cd 1 18110557 543050358 71185  
## + multi 1 11282896 549878019 71263  
## + ads 1 8883646 552277269 71290  
## <none> 561160915 71388  
##   
## Step: AIC=70745.61  
## price ~ ram + trend + speed + premium + screen + hd  
##   
## Df Sum of Sq RSS AIC  
## + ads 1 16662799 489596771 70538  
## + cd 1 14252137 492007433 70569  
## + multi 1 14091100 492168471 70571  
## <none> 506259571 70746  
##   
## Step: AIC=70538.14  
## price ~ ram + trend + speed + premium + screen + hd + ads  
##   
## Df Sum of Sq RSS AIC  
## + multi 1 12705685 476891087 70376  
## + cd 1 9477678 480119093 70418  
## <none> 489596771 70538  
##   
## Step: AIC=70375.56  
## price ~ ram + trend + speed + premium + screen + hd + ads + multi  
##   
## Df Sum of Sq RSS AIC  
## + cd 1 3107211 473783875 70337  
## <none> 476891087 70376  
##   
## Step: AIC=70336.65  
## price ~ ram + trend + speed + premium + screen + hd + ads + multi +   
## cd

summary(stepF)

##   
## Call:  
## lm(formula = price ~ ram + trend + speed + premium + screen +   
## hd + ads + multi + cd, data = c\_prices)  
##   
## 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) 307.98798 60.35341 5.103 3.44e-07 \*\*\*  
## ram 48.25596 1.06608 45.265 < 2e-16 \*\*\*  
## trend -51.84958 0.62871 -82.470 < 2e-16 \*\*\*  
## speed 9.32028 0.18506 50.364 < 2e-16 \*\*\*  
## premium -509.22473 12.34225 -41.259 < 2e-16 \*\*\*  
## screen 123.08904 3.99950 30.776 < 2e-16 \*\*\*  
## hd 0.78178 0.02761 28.311 < 2e-16 \*\*\*  
## ads 0.65729 0.05132 12.809 < 2e-16 \*\*\*  
## multi 104.32382 11.41268 9.141 < 2e-16 \*\*\*  
## cd 60.91671 9.51559 6.402 1.65e-10 \*\*\*  
## ---  
## 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

# Selecting the best combination of the 4 Features / Attributes.

library(leaps) # all subsets regression

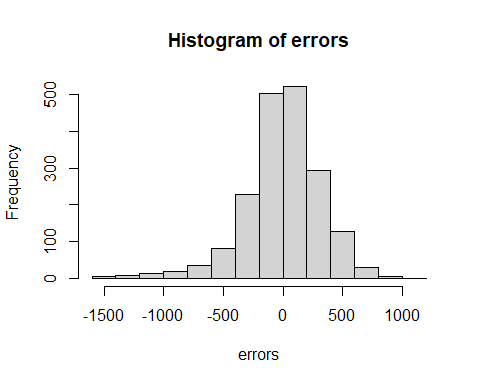
## Warning: package 'leaps' was built under R version 4.1.3

subsets<-regsubsets(price ~ speed + hd + ram + screen + cd + multi + premium + ads + trend, data=c\_prices, nbest=1,)  
sub.sum <- summary(subsets)  
as.data.frame(sub.sum$outmat)

## speed hd ram screen cd multi premium ads trend  
## 1 ( 1 ) \*   
## 2 ( 1 ) \* \*  
## 3 ( 1 ) \* \* \*  
## 4 ( 1 ) \* \* \* \*  
## 5 ( 1 ) \* \* \* \* \*  
## 6 ( 1 ) \* \* \* \* \* \*  
## 7 ( 1 ) \* \* \* \* \* \* \*  
## 8 ( 1 ) \* \* \* \* \* \* \* \*

# Modelling Dataset

rn\_train <- sample(nrow(c\_prices), floor(nrow(c\_prices)\*0.7))  
  
# Modelling with Only Top four Significant Features (Cloumn 4,10,2,8)  
train <- c\_prices[rn\_train,c("price","ram","trend","speed", "premium")]  
test <- c\_prices[-rn\_train,c("price","ram","trend","speed", "premium")]  
  
model\_ulm <- lm(price ~ ram + trend + speed + premium, data=train)   
prediction <- predict(model\_ulm, interval="prediction", newdata =test)  
  
errors <- prediction[,"fit"] - test$price  
hist(errors)



# Calculation of Root Mean Square Error & Percentage of cases that has less than 25% Error

rmse <- sqrt(sum((prediction[,"fit"] - test$price)^2)/nrow(test))  
rel\_change <- 1 - ((test$price - abs(errors)) / test$price)  
pred25 <- table(rel\_change<0.25)["TRUE"] / nrow(test)  
paste("RMSE:", round(rmse,2))

## [1] "RMSE: 324.21"

paste("PRED(25):", round(100\*pred25,2), "%")

## [1] "PRED(25): 90.89 %"

# Predicting the Price of a new Product

library(FNN)

## Warning: package 'FNN' was built under R version 4.1.3

dataset <- rbind(c\_prices, c(0,32,90,8,15,0,0,1,200,2))   
  
dataset.numeric <- as.data.frame(dataset)  
prediction <- knn.reg(dataset.numeric[1:nrow(c\_prices),-1],   
 test = dataset.numeric[nrow(c\_prices)+1,-1],  
 dataset.numeric[1:nrow(c\_prices),]$price, k = 7 , algorithm="kd\_tree")   
  
paste("New Computer Price: $", prediction$pred)

## [1] "New Computer Price: $ 1601"