R Notebook

# Packages and libraries

# install.packages("moments")  
# install.packages("ggplot2")  
# install.packages("corrplot")  
# install.packages("tidyr")  
# install.packages("dplyr")  
# install.packages("ggridges")  
# install.packages("mlbench")  
# install.packages("lattice")  
# install.packages("caret")  
# install.packages("broom")  
# install.packages("C50")  
# install.packages("rpart")  
# install.packages("MASS")  
# install.packages("leaps")  
library(moments)  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.4

library("corrplot")

## Warning: package 'corrplot' was built under R version 3.4.4

## corrplot 0.84 loaded

library(tidyr)

## Warning: package 'tidyr' was built under R version 3.4.4

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.4

##   
## 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

library(ggridges)

## Warning: package 'ggridges' was built under R version 3.4.4

library(mlbench)

## Warning: package 'mlbench' was built under R version 3.4.4

library(caret)

## Warning: package 'caret' was built under R version 3.4.4

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 3.4.4

library("C50")

## Warning: package 'C50' was built under R version 3.4.4

library(rpart)  
library(MASS)

## Warning: package 'MASS' was built under R version 3.4.4

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(leaps)

## Warning: package 'leaps' was built under R version 3.4.4

# Load Data   
ccfraud <- read.csv("C:/Users/Kiran Kandhola/Documents/creditcardfraud/creditcard.csv",stringsAsFactors = FALSE)

# Check the head and data types of the attributes  
cc <- ccfraud ## Make a copy of the data  
head(cc) ## view first few records of the dataset

## Time V1 V2 V3 V4 V5 V6  
## 1 0 -1.3598071 -0.07278117 2.5363467 1.3781552 -0.33832077 0.46238778  
## 2 0 1.1918571 0.26615071 0.1664801 0.4481541 0.06001765 -0.08236081  
## 3 1 -1.3583541 -1.34016307 1.7732093 0.3797796 -0.50319813 1.80049938  
## 4 1 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888 1.24720317  
## 5 2 -1.1582331 0.87773675 1.5487178 0.4030339 -0.40719338 0.09592146  
## 6 2 -0.4259659 0.96052304 1.1411093 -0.1682521 0.42098688 -0.02972755  
## V7 V8 V9 V10 V11 V12  
## 1 0.23959855 0.09869790 0.3637870 0.09079417 -0.5515995 -0.61780086  
## 2 -0.07880298 0.08510165 -0.2554251 -0.16697441 1.6127267 1.06523531  
## 3 0.79146096 0.24767579 -1.5146543 0.20764287 0.6245015 0.06608369  
## 4 0.23760894 0.37743587 -1.3870241 -0.05495192 -0.2264873 0.17822823  
## 5 0.59294075 -0.27053268 0.8177393 0.75307443 -0.8228429 0.53819555  
## 6 0.47620095 0.26031433 -0.5686714 -0.37140720 1.3412620 0.35989384  
## V13 V14 V15 V16 V17 V18  
## 1 -0.9913898 -0.3111694 1.4681770 -0.4704005 0.20797124 0.02579058  
## 2 0.4890950 -0.1437723 0.6355581 0.4639170 -0.11480466 -0.18336127  
## 3 0.7172927 -0.1659459 2.3458649 -2.8900832 1.10996938 -0.12135931  
## 4 0.5077569 -0.2879237 -0.6314181 -1.0596472 -0.68409279 1.96577500  
## 5 1.3458516 -1.1196698 0.1751211 -0.4514492 -0.23703324 -0.03819479  
## 6 -0.3580907 -0.1371337 0.5176168 0.4017259 -0.05813282 0.06865315  
## V19 V20 V21 V22 V23  
## 1 0.40399296 0.25141210 -0.018306778 0.277837576 -0.11047391  
## 2 -0.14578304 -0.06908314 -0.225775248 -0.638671953 0.10128802  
## 3 -2.26185710 0.52497973 0.247998153 0.771679402 0.90941226  
## 4 -1.23262197 -0.20803778 -0.108300452 0.005273597 -0.19032052  
## 5 0.80348692 0.40854236 -0.009430697 0.798278495 -0.13745808  
## 6 -0.03319379 0.08496767 -0.208253515 -0.559824796 -0.02639767  
## V24 V25 V26 V27 V28 Amount Class  
## 1 0.06692807 0.1285394 -0.1891148 0.133558377 -0.02105305 149.62 0  
## 2 -0.33984648 0.1671704 0.1258945 -0.008983099 0.01472417 2.69 0  
## 3 -0.68928096 -0.3276418 -0.1390966 -0.055352794 -0.05975184 378.66 0  
## 4 -1.17557533 0.6473760 -0.2219288 0.062722849 0.06145763 123.50 0  
## 5 0.14126698 -0.2060096 0.5022922 0.219422230 0.21515315 69.99 0  
## 6 -0.37142658 -0.2327938 0.1059148 0.253844225 0.08108026 3.67 0

str(cc) ## Check the type of all the attributeslevels(cc\_trainl$Class)

## 'data.frame': 284807 obs. of 31 variables:  
## $ Time : num 0 0 1 1 2 2 4 7 7 9 ...  
## $ V1 : num -1.36 1.192 -1.358 -0.966 -1.158 ...  
## $ V2 : num -0.0728 0.2662 -1.3402 -0.1852 0.8777 ...  
## $ V3 : num 2.536 0.166 1.773 1.793 1.549 ...  
## $ V4 : num 1.378 0.448 0.38 -0.863 0.403 ...  
## $ V5 : num -0.3383 0.06 -0.5032 -0.0103 -0.4072 ...  
## $ V6 : num 0.4624 -0.0824 1.8005 1.2472 0.0959 ...  
## $ V7 : num 0.2396 -0.0788 0.7915 0.2376 0.5929 ...  
## $ V8 : num 0.0987 0.0851 0.2477 0.3774 -0.2705 ...  
## $ V9 : num 0.364 -0.255 -1.515 -1.387 0.818 ...  
## $ V10 : num 0.0908 -0.167 0.2076 -0.055 0.7531 ...  
## $ V11 : num -0.552 1.613 0.625 -0.226 -0.823 ...  
## $ V12 : num -0.6178 1.0652 0.0661 0.1782 0.5382 ...  
## $ V13 : num -0.991 0.489 0.717 0.508 1.346 ...  
## $ V14 : num -0.311 -0.144 -0.166 -0.288 -1.12 ...  
## $ V15 : num 1.468 0.636 2.346 -0.631 0.175 ...  
## $ V16 : num -0.47 0.464 -2.89 -1.06 -0.451 ...  
## $ V17 : num 0.208 -0.115 1.11 -0.684 -0.237 ...  
## $ V18 : num 0.0258 -0.1834 -0.1214 1.9658 -0.0382 ...  
## $ V19 : num 0.404 -0.146 -2.262 -1.233 0.803 ...  
## $ V20 : num 0.2514 -0.0691 0.525 -0.208 0.4085 ...  
## $ V21 : num -0.01831 -0.22578 0.248 -0.1083 -0.00943 ...  
## $ V22 : num 0.27784 -0.63867 0.77168 0.00527 0.79828 ...  
## $ V23 : num -0.11 0.101 0.909 -0.19 -0.137 ...  
## $ V24 : num 0.0669 -0.3398 -0.6893 -1.1756 0.1413 ...  
## $ V25 : num 0.129 0.167 -0.328 0.647 -0.206 ...  
## $ V26 : num -0.189 0.126 -0.139 -0.222 0.502 ...  
## $ V27 : num 0.13356 -0.00898 -0.05535 0.06272 0.21942 ...  
## $ V28 : num -0.0211 0.0147 -0.0598 0.0615 0.2152 ...  
## $ Amount: num 149.62 2.69 378.66 123.5 69.99 ...  
## $ Class : int 0 0 0 0 0 0 0 0 0 0 ...

# Missing Values  
navalue <- sum(is.na(cc)) # Find missing values  
navalue # There is no missing value in the data

## [1] 0

cc <- ccfraud # Make a copy of the data  
cc$Class <- as.factor(cc$Class) # Convert the class to factors  
cc$Class <- factor(cc$Class, levels = c("1", "0")) # Change the order of levels   
ccfd <- cc   
levels(ccfd$Class) <- c("Fraud", "Genuine") # Change the name of levels to Fraud and Genuine  
summary(ccfd)

## Time V1 V2   
## Min. : 0 Min. :-56.40751 Min. :-72.71573   
## 1st Qu.: 54202 1st Qu.: -0.92037 1st Qu.: -0.59855   
## Median : 84692 Median : 0.01811 Median : 0.06549   
## Mean : 94814 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.:139321 3rd Qu.: 1.31564 3rd Qu.: 0.80372   
## Max. :172792 Max. : 2.45493 Max. : 22.05773   
## V3 V4 V5   
## Min. :-48.3256 Min. :-5.68317 Min. :-113.74331   
## 1st Qu.: -0.8904 1st Qu.:-0.84864 1st Qu.: -0.69160   
## Median : 0.1799 Median :-0.01985 Median : -0.05434   
## Mean : 0.0000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 1.0272 3rd Qu.: 0.74334 3rd Qu.: 0.61193   
## Max. : 9.3826 Max. :16.87534 Max. : 34.80167   
## V6 V7 V8   
## Min. :-26.1605 Min. :-43.5572 Min. :-73.21672   
## 1st Qu.: -0.7683 1st Qu.: -0.5541 1st Qu.: -0.20863   
## Median : -0.2742 Median : 0.0401 Median : 0.02236   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000   
## 3rd Qu.: 0.3986 3rd Qu.: 0.5704 3rd Qu.: 0.32735   
## Max. : 73.3016 Max. :120.5895 Max. : 20.00721   
## V9 V10 V11   
## Min. :-13.43407 Min. :-24.58826 Min. :-4.79747   
## 1st Qu.: -0.64310 1st Qu.: -0.53543 1st Qu.:-0.76249   
## Median : -0.05143 Median : -0.09292 Median :-0.03276   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 0.59714 3rd Qu.: 0.45392 3rd Qu.: 0.73959   
## Max. : 15.59500 Max. : 23.74514 Max. :12.01891   
## V12 V13 V14   
## Min. :-18.6837 Min. :-5.79188 Min. :-19.2143   
## 1st Qu.: -0.4056 1st Qu.:-0.64854 1st Qu.: -0.4256   
## Median : 0.1400 Median :-0.01357 Median : 0.0506   
## Mean : 0.0000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.6182 3rd Qu.: 0.66251 3rd Qu.: 0.4931   
## Max. : 7.8484 Max. : 7.12688 Max. : 10.5268   
## V15 V16 V17   
## Min. :-4.49894 Min. :-14.12985 Min. :-25.16280   
## 1st Qu.:-0.58288 1st Qu.: -0.46804 1st Qu.: -0.48375   
## Median : 0.04807 Median : 0.06641 Median : -0.06568   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 0.64882 3rd Qu.: 0.52330 3rd Qu.: 0.39968   
## Max. : 8.87774 Max. : 17.31511 Max. : 9.25353   
## V18 V19 V20   
## Min. :-9.498746 Min. :-7.213527 Min. :-54.49772   
## 1st Qu.:-0.498850 1st Qu.:-0.456299 1st Qu.: -0.21172   
## Median :-0.003636 Median : 0.003735 Median : -0.06248   
## Mean : 0.000000 Mean : 0.000000 Mean : 0.00000   
## 3rd Qu.: 0.500807 3rd Qu.: 0.458949 3rd Qu.: 0.13304   
## Max. : 5.041069 Max. : 5.591971 Max. : 39.42090   
## V21 V22 V23   
## Min. :-34.83038 Min. :-10.933144 Min. :-44.80774   
## 1st Qu.: -0.22839 1st Qu.: -0.542350 1st Qu.: -0.16185   
## Median : -0.02945 Median : 0.006782 Median : -0.01119   
## Mean : 0.00000 Mean : 0.000000 Mean : 0.00000   
## 3rd Qu.: 0.18638 3rd Qu.: 0.528554 3rd Qu.: 0.14764   
## Max. : 27.20284 Max. : 10.503090 Max. : 22.52841   
## V24 V25 V26   
## Min. :-2.83663 Min. :-10.29540 Min. :-2.60455   
## 1st Qu.:-0.35459 1st Qu.: -0.31715 1st Qu.:-0.32698   
## Median : 0.04098 Median : 0.01659 Median :-0.05214   
## Mean : 0.00000 Mean : 0.00000 Mean : 0.00000   
## 3rd Qu.: 0.43953 3rd Qu.: 0.35072 3rd Qu.: 0.24095   
## Max. : 4.58455 Max. : 7.51959 Max. : 3.51735   
## V27 V28 Amount   
## Min. :-22.565679 Min. :-15.43008 Min. : 0.00   
## 1st Qu.: -0.070840 1st Qu.: -0.05296 1st Qu.: 5.60   
## Median : 0.001342 Median : 0.01124 Median : 22.00   
## Mean : 0.000000 Mean : 0.00000 Mean : 88.35   
## 3rd Qu.: 0.091045 3rd Qu.: 0.07828 3rd Qu.: 77.17   
## Max. : 31.612198 Max. : 33.84781 Max. :25691.16   
## Class   
## Fraud : 492   
## Genuine:284315   
##   
##   
##   
##

# Summary Statistics of the data  
prop.table(table(ccfd$Class)) # Table of Fraud and Genuine cases

##   
## Fraud Genuine   
## 0.001727486 0.998272514

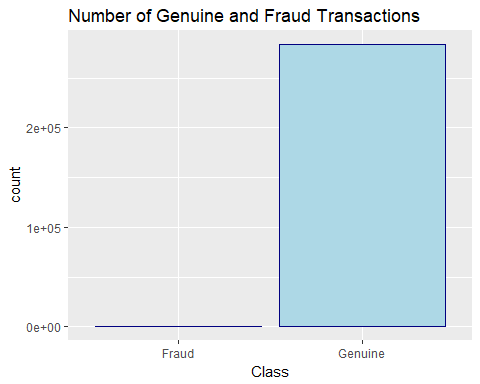
fraudpercent <- sum(as.numeric(ccfraud$Class))/nrow(ccfraud)   
sprintf('Percentage of fraudulent transactions in the data set %f', fraudpercent\*100)

## [1] "Percentage of fraudulent transactions in the data set 0.172749"

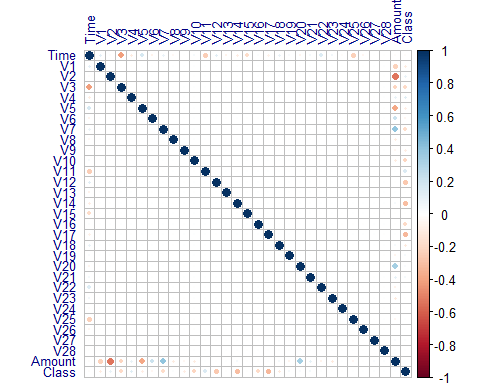
# Check skewness of the data  
skewness(ccfraud)

## Time V1 V2 V3 V4 V5   
## -0.03556743 -3.28065002 -4.62484122 -2.24014364 0.67628854 -2.42588872   
## V6 V7 V8 V9 V10 V11   
## 1.82657104 2.55389397 -8.52189931 0.55467685 1.18713434 0.35650398   
## V12 V13 V14 V15 V16 V17   
## -2.27838894 0.06523311 -1.99516533 -0.30842136 -1.10096048 -3.84489422   
## V18 V19 V20 V21 V22 V23   
## -0.25987890 0.10919118 -2.03714457 3.59297227 -0.21325650 -5.87510940   
## V24 V25 V26 V27 V28 Amount   
## -0.55249639 -0.41579040 0.57668958 -1.17020278 11.19203225 16.97763504   
## Class   
## 23.99745292

## Visulaization of class imbalance  
ggplot(ccfd, aes(x = Class)) + geom\_bar(fill= "lightblue", colour = "navyblue") + ggtitle("Number of Genuine and Fraud Transactions")

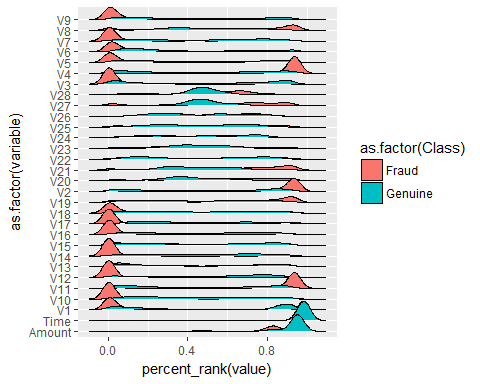


# correlations between the attributes and the "Class"  
correlations <- cor(ccfraud)  
corrplot(correlations, method = "circle", type = "full", number.cex = .9, tl.cex=0.8, tl.col = "navyblue")



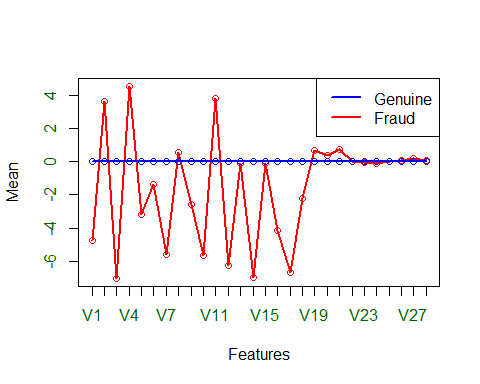
ccfd %>%  
 gather(variable, value, -Class) %>%  
 ggplot(aes(y = as.factor(variable),   
 fill = as.factor(Class),   
 x = percent\_rank(value)))+scale\_color\_hue(l=40, c=35) + geom\_density\_ridges()

## Picking joint bandwidth of 0.0309



# For aclassifier to work well we have a strong initial assumption: that the distribution of variables for normal transactions is different from the distribution for fraudulent ones. Let's make some plots to verify this. Variables were transformed to a [0,1] interval for plotting.  
# We can see that distributions of variables for fraudulent transactions are very different then from normal ones, except for the Time variable, which seems to have the exact same distribution

# Visualization of the mean values of all the features for fraudulent and genuine transactions  
rownames(ccfd) <- 1:nrow(ccfd)  
Genuine <- ccfd[ccfd$Class == "Genuine",]  
Fraud <- ccfd[ccfd$Class == "Fraud",]  
mugenuine <- apply(Genuine[, -c(1, 30, 31)], 2, mean)  
mufraud <- apply(Fraud[, -c(1, 30, 31)], 2, mean)  
plot(mufraud, col = "red",xaxt = "n",xlab = "Features", ylab = "Mean", col.axis = "darkgreen")  
lines(mufraud, col = "red", lwd = 2)  
points(mugenuine, col = "blue")  
lines(mugenuine, col = "blue", lwd = 2)  
legend("topright", legend = c("Genuine", "Fraud"), lty = c(1,1), col = c("blue", "red"), lwd = c(2,2))  
a <- c("V")  
b <- c(1:28)  
xlabels<- paste(a,b,sep="")  
axis(side = 1,at = 1:28, labels = xlabels, col.axis="darkgreen")



# Remove Redundant Features with an absolute correlation of 0.75 or higher.  
set.seed(7)  
correlationMatrix <- cor(ccfraud[ , -c(31)]) # calculate correlation matrix  
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.75)   
print(highlyCorrelated) # print indexes of attributes that are highly correlated to other remaining variables

## integer(0)

# none of the attributes are highly correlated to each other as all attributes are already PCA transformed

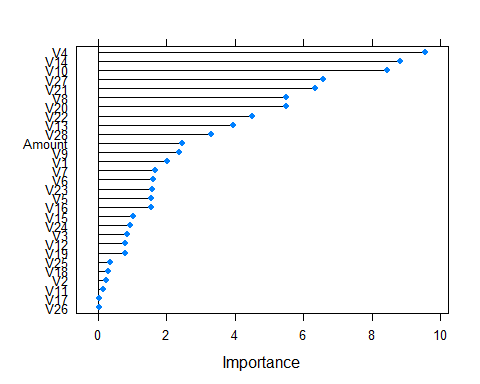
# Rank of features by importance using glm  
set.seed(7)  
control <- trainControl(method="repeatedcv", number=10, repeats=3) # prepare training scheme  
model <- train(Class~., data=cc[ , -c(1)], method="glm", trControl=control) # train the model

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

importance <- varImp(model, scale=FALSE) # estimate variable importance  
print(importance) # summarize importance

## glm variable importance  
##   
## only 20 most important variables shown (out of 29)  
##   
## Overall  
## V4 9.5782  
## V14 8.8431  
## V10 8.4389  
## V27 6.5678  
## V21 6.3343  
## V8 5.5106  
## V20 5.4877  
## V22 4.5144  
## V13 3.9367  
## V28 3.2950  
## Amount 2.4584  
## V9 2.3748  
## V1 2.0160  
## V7 1.6742  
## V6 1.6212  
## V23 1.5680  
## V5 1.5594  
## V16 1.5508  
## V15 1.0159  
## V24 0.9282

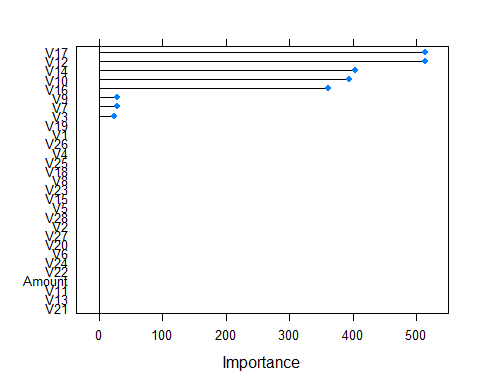
plot(importance) # plot importance



# Rank of features by importance using rpart  
set.seed(7)   
control <- trainControl(method="cv", number=10) # prepare training scheme  
model <- train(Class~., data=cc[ , -c(1)], method="rpart", trControl=control) # train the model using logistic  
importance <- varImp(model, scale=FALSE) # estimate variable importance  
print(importance) # summarize and plot importance

## rpart variable importance  
##   
## only 20 most important variables shown (out of 29)  
##   
## Overall  
## V17 514.24  
## V12 513.36  
## V14 403.76  
## V10 394.07  
## V16 361.67  
## V9 29.54  
## V7 29.00  
## V3 24.39  
## V2 0.00  
## V11 0.00  
## V4 0.00  
## V18 0.00  
## V5 0.00  
## V26 0.00  
## V15 0.00  
## V21 0.00  
## V22 0.00  
## V23 0.00  
## V27 0.00  
## V24 0.00

plot(importance)



# Rank of features using C5.0  
#set.seed(7)   
#control <- trainControl(method="cv", number=10) # prepare training scheme  
#model <- train(Class~., data=cc[ ,-c(1)], method="C5.0", trControl=control, importance = TRUE)  
#importance <- varImp(model, scale=FALSE) # estimate variable importance  
#print(importance) # summarize and plot importance  
#plot(importance)

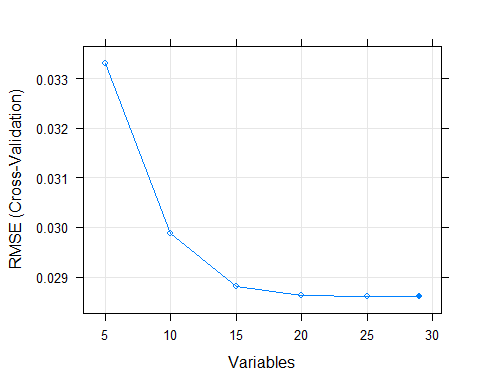
# Recursive feature elimination  
subsets <- c(5,10,15,20,25,30)  
set.seed(7)   
ctrl <- rfeControl(functions=lmFuncs, method="cv", number=10) # define the control  
lmProfile <- rfe(ccfraud[,2:30], ccfraud[,31],sizes = subsets, rfeControl = ctrl)  
print(lmProfile) # summarize the results

##   
## Recursive feature selection  
##   
## Outer resampling method: Cross-Validated (10 fold)   
##   
## Resampling performance over subset size:  
##   
## Variables RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 5 0.03331 0.3538 0.011130 0.003216 0.02685 0.0001805  
## 10 0.02988 0.4828 0.007796 0.003150 0.03243 0.0001991  
## 15 0.02880 0.5211 0.004488 0.003302 0.03770 0.0005453  
## 20 0.02862 0.5272 0.003412 0.003265 0.03876 0.0001697  
## 25 0.02861 0.5276 0.003342 0.003267 0.03887 0.0001747  
## 29 0.02860 0.5279 0.003373 0.003268 0.03882 0.0001719  
## Selected  
##   
##   
##   
##   
##   
## \*  
##   
## The top 5 variables (out of 29):  
## V17, V14, V12, V16, V10

predictors(lmProfile) # list the chosen features

## [1] "V17" "V14" "V12" "V16" "V10" "V7" "V11"   
## [8] "V18" "V3" "V4" "V9" "V2" "V5" "V21"   
## [15] "V1" "V19" "V27" "V6" "V28" "V8" "V24"   
## [22] "V25" "V26" "V20" "V22" "V13" "V15" "V23"   
## [29] "Amount"

plot(lmProfile, type=c("g", "o")) # plot the results



lmProfile

##   
## Recursive feature selection  
##   
## Outer resampling method: Cross-Validated (10 fold)   
##   
## Resampling performance over subset size:  
##   
## Variables RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 5 0.03331 0.3538 0.011130 0.003216 0.02685 0.0001805  
## 10 0.02988 0.4828 0.007796 0.003150 0.03243 0.0001991  
## 15 0.02880 0.5211 0.004488 0.003302 0.03770 0.0005453  
## 20 0.02862 0.5272 0.003412 0.003265 0.03876 0.0001697  
## 25 0.02861 0.5276 0.003342 0.003267 0.03887 0.0001747  
## 29 0.02860 0.5279 0.003373 0.003268 0.03882 0.0001719  
## Selected  
##   
##   
##   
##   
##   
## \*  
##   
## The top 5 variables (out of 29):  
## V17, V14, V12, V16, V10

# The results suggest that taking all the attributes will give better results. However there should not be much difference in the results if we choose a subset of 20 attributes or 25 or 30.

# Step Forward Selection

full <- lm(Class ~. ,data = ccfraud[ ,-c(1)])  
null <- lm(Class ~ 1, data = ccfraud[ ,-c(1)])  
stepF <- stepAIC(null,scope = list(lower = null, upper = full),direction = "forward", trace = FALSE)  
stepF$anova

## Stepwise Model Path   
## Analysis of Deviance Table  
##   
## Initial Model:  
## Class ~ 1  
##   
## Final Model:  
## Class ~ V17 + V14 + V12 + V10 + V16 + V3 + V7 + V11 + V4 + V18 +   
## V1 + V9 + V5 + V2 + V6 + V21 + V19 + V20 + V8 + V27 + Amount +   
## V28 + V24 + V25 + V13 + V26 + V15 + V22 + V23  
##   
##   
## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 284806 491.1501 -1812173  
## 2 + V17 1 52.351631347 284805 438.7984 -1844271  
## 3 + V14 1 44.956286702 284804 393.8422 -1875054  
## 4 + V12 1 33.353349730 284803 360.4888 -1900255  
## 5 + V10 1 23.102821077 284802 337.3860 -1919116  
## 6 + V16 1 18.971926643 284801 318.4141 -1935597  
## 7 + V3 1 18.287423416 284800 300.1266 -1952441  
## 8 + V7 1 17.222192710 284799 282.9044 -1969270  
## 9 + V11 1 11.780954298 284798 271.1235 -1981382  
## 10 + V4 1 8.746514315 284797 262.3770 -1990720  
## 11 + V18 1 6.104485563 284796 256.2725 -1997422  
## 12 + V1 1 5.044737475 284795 251.2278 -2003083  
## 13 + V9 1 4.691307350 284794 246.5364 -2008449  
## 14 + V5 1 4.430231390 284793 242.1062 -2013612  
## 15 + V2 1 4.093056967 284792 238.0132 -2018466  
## 16 + V6 1 0.935506064 284791 237.0777 -2019586  
## 17 + V21 1 0.802166606 284790 236.2755 -2020549  
## 18 + V19 1 0.594221848 284789 235.6813 -2021264  
## 19 + V20 1 0.198238547 284788 235.4830 -2021502  
## 20 + V8 1 0.194014374 284787 235.2890 -2021734  
## 21 + V27 1 0.151788381 284786 235.1372 -2021916  
## 22 + Amount 1 0.065459776 284785 235.0718 -2021993  
## 23 + V28 1 0.041298166 284784 235.0305 -2022042  
## 24 + V24 1 0.026918940 284783 235.0035 -2022072  
## 25 + V25 1 0.012619347 284782 234.9909 -2022085  
## 26 + V13 1 0.011183284 284781 234.9797 -2022097  
## 27 + V26 1 0.010295564 284780 234.9694 -2022107  
## 28 + V15 1 0.008291879 284779 234.9612 -2022116  
## 29 + V22 1 0.005542568 284778 234.9556 -2022120  
## 30 + V23 1 0.002005653 284777 234.9536 -2022121

# Step Backward Selection

full <- lm(Class ~. ,data = ccfraud[ ,-c(1)])  
stepB <- stepAIC(full,direction = "backward", trace = FALSE)  
stepB$anova

## Stepwise Model Path   
## Analysis of Deviance Table  
##   
## Initial Model:  
## Class ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +   
## V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +   
## V22 + V23 + V24 + V25 + V26 + V27 + V28 + Amount  
##   
## Final Model:  
## Class ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +   
## V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +   
## V22 + V23 + V24 + V25 + V26 + V27 + V28 + Amount  
##   
##   
## Step Df Deviance Resid. Df Resid. Dev AIC  
## 1 284777 234.9536 -2022121

# Selection of best subsets

subsets <- regsubsets(Class ~. ,data = ccfd, nbest = 1)  
sub.sum <- summary(subsets)  
as.data.frame(sub.sum$outmat)

## Time V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17  
## 1 ( 1 ) \*  
## 2 ( 1 ) \* \*  
## 3 ( 1 ) \* \* \*  
## 4 ( 1 ) \* \* \* \*  
## 5 ( 1 ) \* \* \* \* \*  
## 6 ( 1 ) \* \* \* \* \* \*  
## 7 ( 1 ) \* \* \* \* \* \* \*  
## 8 ( 1 ) \* \* \* \* \* \* \* \*  
## V18 V19 V20 V21 V22 V23 V24 V25 V26 V27 V28 Amount  
## 1 ( 1 )   
## 2 ( 1 )   
## 3 ( 1 )   
## 4 ( 1 )   
## 5 ( 1 )   
## 6 ( 1 )   
## 7 ( 1 )   
## 8 ( 1 )

## the best 8 attributes are V17, V14, V12, V10, V16, V3, V7, V11