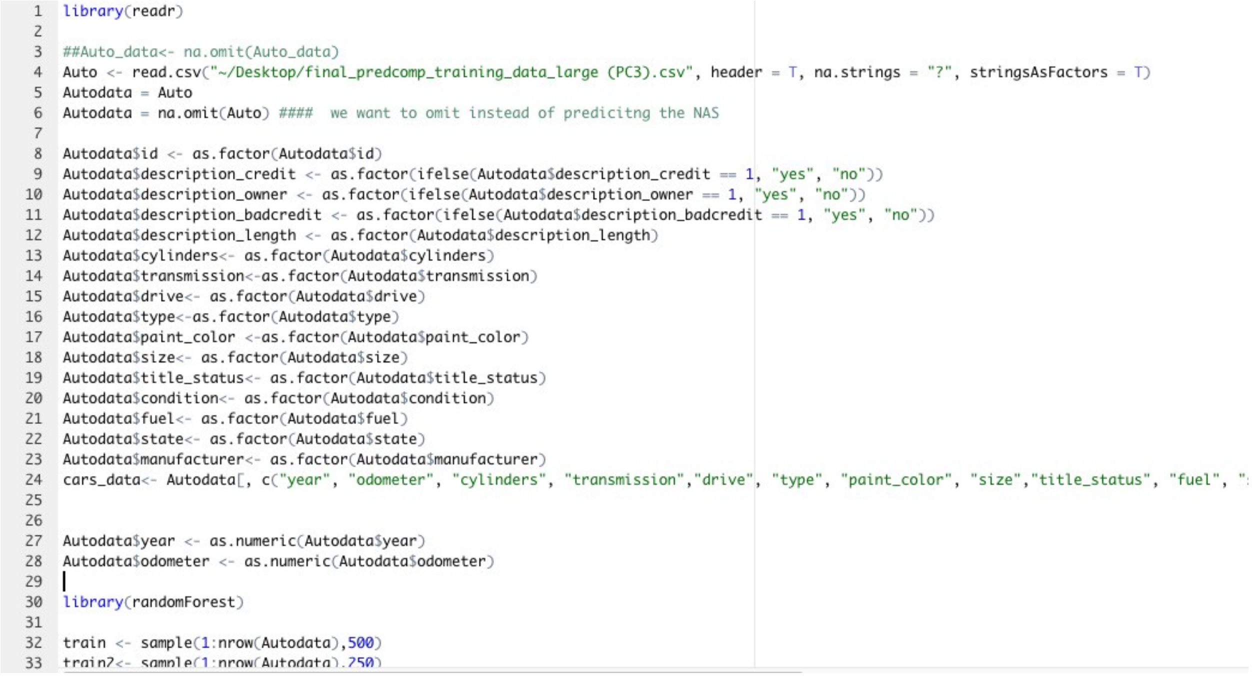
PREDICTION COMPETITION 4

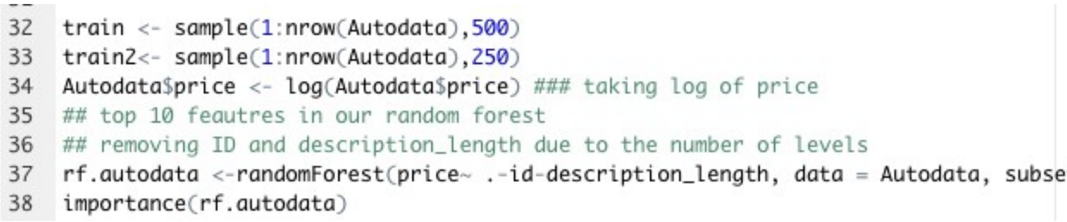
Q1

Loading data, creating factors and omitting data :



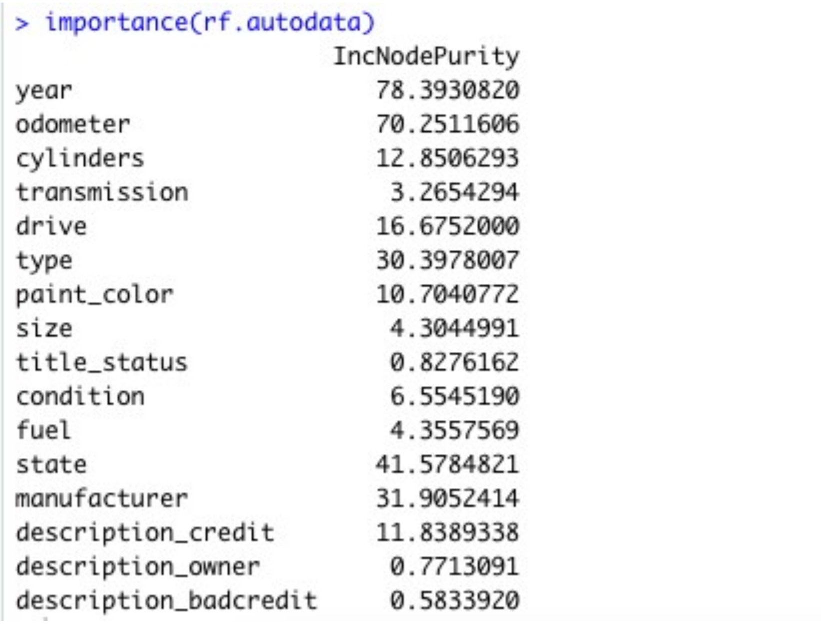
```
library(readr)

##Auto_data<- na.omit(Auto_data)
Auto <- read.csv("~/Desktop/final_predcomp_training_data_large (PC3).csv", header = T, na.strings = "?", stringsAsFactors = T)
Autodata = Auto
Autodata = na.omit(Auto) ####  we want to omit instead of predicitng the NAS

Autodata$id <- as.factor(Autodata$id)
Autodata$description_credit <- as.factor(ifelse(Autodata$description_credit == 1, "yes", "no"))
Autodata$description_owner <- as.factor(ifelse(Autodata$description_owner == 1, "yes", "no"))
Autodata$description_badcredit <- as.factor(ifelse(Autodata$description_badcredit == 1, "yes", "no"))
Autodata$description_length <- as.factor(Autodata$description_length)
Autodata$cylinders<- as.factor(Autodata$cylinders)
Autodata$transmission<-as.factor(Autodata$transmission)
Autodata$drive<- as.factor(Autodata$drive)
Autodata$type<-as.factor(Autodata$type)
Autodata$paint_color <-as.factor(Autodata$paint_color)
Autodata$size<- as.factor(Autodata$size)
Autodata$title_status<- as.factor(Autodata$title_status)
Autodata$condition<- as.factor(Autodata$condition)
Autodata$fuel<- as.factor(Autodata$fuel)
Autodata$state<- as.factor(Autodata$state)
Autodata$manufacturer<- as.factor(Autodata$manufacturer)
cars_data<- Autodata[, c("year", "odometer", "cylinders", "transmission","drive", "type", "paint_color", "size","title_status", "fuel", "


Autodata$year <- as.numeric(Autodata$year)
Autodata$odometer <- as.numeric(Autodata$odometer)

library(randomForest)

train <- sample(1:nrow(Autodata),500)
train2<- sample(1:nrow(Autodata),250)
```

Due to the huge number of categories and levels in ID and description_length, I made a random forest of all the data except the two.



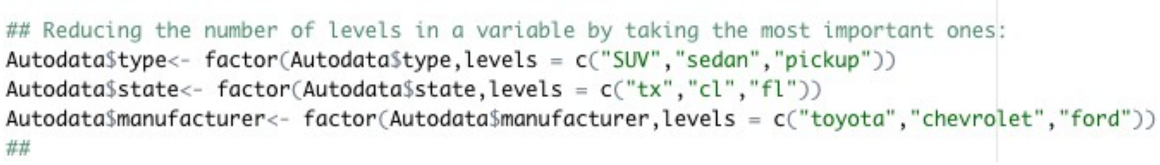
```
train <- sample(1:nrow(Autodata),500)
train2<- sample(1:nrow(Autodata),250)
Autodata$price <- log(Autodata$price) ### taking log of price
## top 10 feautres in our random forest
## removing ID and description_length due to the number of levels
rf.autodata <-randomForest(price~ .-id-description_length, data = Autodata, subset =train)
importance(rf.autodata)
```

After creating a random forest of the data, we can see the importance of the various variables:



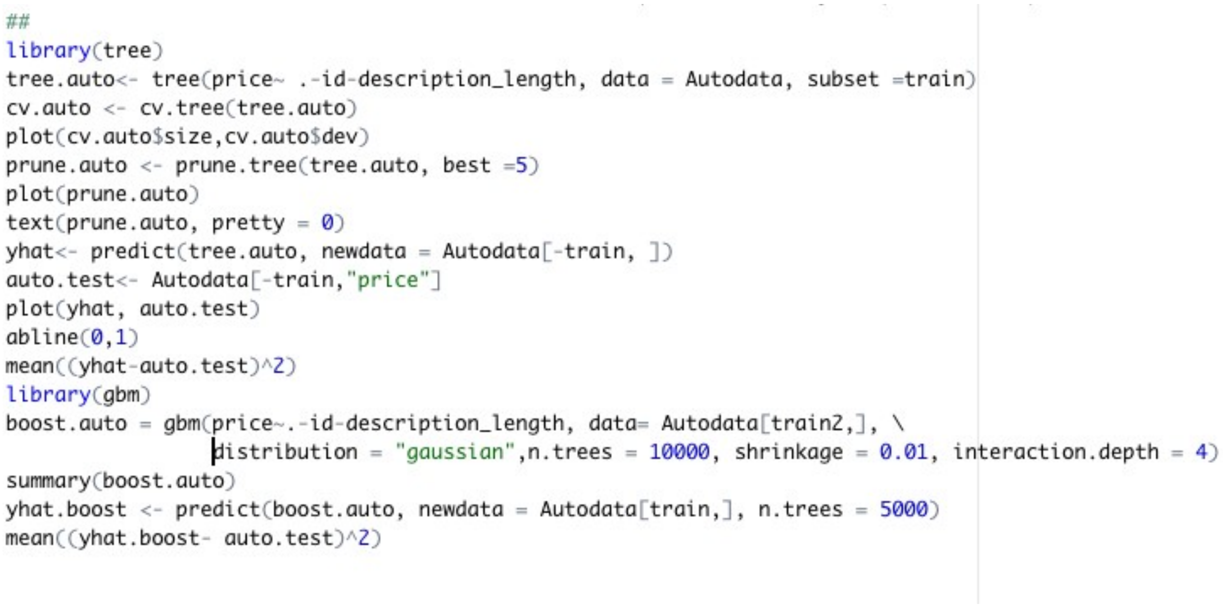
```
> importance(rf.autodata)
                      IncNodePurity
year                     78.3930820
odometer                 70.2511606
cylinders                12.8506293
transmission              3.2654294
drive                    16.6752000
type                     30.3978007
paint_color              10.7040772
size                      4.3044991
title_status              0.8276162
condition                 6.5545190
fuel                      4.3557569
state                    41.5784821
manufacturer             31.9052414
description_credit       11.8389338
description_owner         0.7713091
description_badcredit     0.5833920
```

Reducing the number of levels in these following 3 variables in order to make a tree as we cannot use more than 32 levels. For this I used the summary method in order to see the levels which were the most popular.



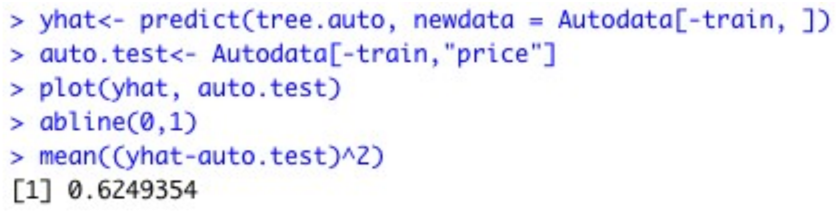
```
## Reducing the number of levels in a variable by taking the most important ones:
Autodata$type<- factor(Autodata$type,levels = c("SUV","sedan","pickup"))
Autodata$state<- factor(Autodata$state,levels = c("tx","cl","fl"))
Autodata$manufacturer<- factor(Autodata$manufacturer,levels = c("toyota","chevrolet","ford"))
##
```

Creating a tree using all the variables as in the random forest(except ID and description_length):



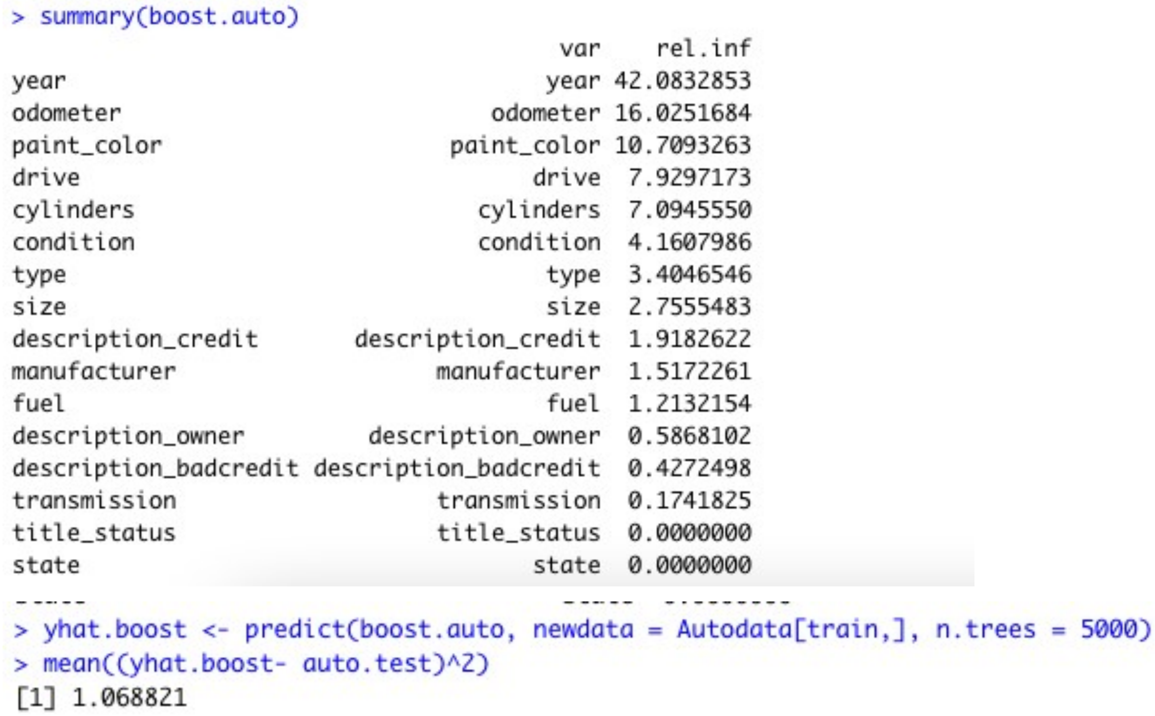
```
##
library(tree)
tree.auto<- tree(price~ .-id-description_length, data = Autodata, subset =train)
cv.auto <- cv.tree(tree.auto)
plot(cv.auto$size,cv.auto$dev)
prune.auto <- prune.tree(tree.auto, best =5)
plot(prune.auto)
text(prune.auto, pretty = 0)
yhat<- predict(tree.auto, newdata = Autodata[-train, ])
auto.test<- Autodata[-train,"price"]
plot(yhat, auto.test)
abline(0,1)
mean((yhat-auto.test)^2)
library(gbm)
boost.auto = gbm(price~.-id-description_length, data= Autodata[train2,], \
                distribution = "gaussian",n.trees = 10000, shrinkage = 0.01, interaction.depth = 4)
summary(boost.auto)
yhat.boost <- predict(boost.auto, newdata = Autodata[train,], n.trees = 5000)
mean((yhat.boost- auto.test)^2)
```

For the mean of the data, we get :



```
> yhat<- predict(tree.auto, newdata = Autodata[-train, ])
> auto.test<- Autodata[-train,"price"]
> plot(yhat, auto.test)
> abline(0,1)
> mean((yhat-auto.test)^2)
[1] 0.6249354
```

Upon boosting the data, I got the following :



```
> summary(boost.auto)
                                        var    rel.inf
year                                   year 42.0832853
odometer                           odometer 16.0251684
paint_color                     paint_color 10.7093263
drive                                 drive  7.9297173
cylinders                         cylinders  7.0945550
condition                         condition  4.1607986
type                                   type  3.4046546
size                                   size  2.7555483
description_credit       description_credit  1.9182622
manufacturer                   manufacturer  1.5172261
fuel                                   fuel  1.2132154
description_owner         description_owner  0.5868102
description_badcredit description_badcredit  0.4272498
transmission                   transmission  0.1741825
title_status                   title_status  0.0000000
state                                 state  0.0000000
> yhat.boost <- predict(boost.auto, newdata = Autodata[train,], n.trees = 5000)
> mean((yhat.boost- auto.test)^2)
[1] 1.068821
```