hw4

Q1. Utilize either of the two the train- ing data sets to train a model that predicts the logarithm of car prices.

Step 1. Import data, imputation for missing values, and set categorical data

```
dat <- read.csv("C:/Users/linzh/Desktop/ECON 626/hw3/final_predcomp_training_data_small.csv")
dat$lprice <- log(dat$price)

#categorical variable: fuel
unique(dat$fuel)
```

```
## [1] "diesel"   "gas"      "hybrid"   "other"    ""         "electric"
```

```
mapping <- c("gas" = 0, "diesel" = 1, "hybrid" = 2, "electric" = 3, "other" = 4)
dat$fuel <- mapping[dat$fuel]
dat$fuel[is.na(dat$fuel)] <- 9

#categorical variable: transmission
unique(dat$transmission)
```

```
## [1] "automatic" "other"     "manual"    ""
```

```
mapping <- c("automatic" = 0, "manual" = 1, "other" = 2)
dat$transmission <- mapping[dat$transmission]
dat$transmission[is.na(dat$transmission)] <- 9

#categorical variable: condition
unique(dat$condition)
```

```
## [1] ""          "good"      "excellent" "like new"  "fair"      "new"
## [7] "salvage"
```

```
mapping <- c("new" = 0, "excellent" = 1, "good" = 2, "like new" = 3, "fair" = 4, "salvage" = 5)
dat$condition <- mapping[dat$condition]
dat$condition[is.na(dat$condition)] <- 9

# fill in na in year and odometer
dat$year[is.na(dat$year)] <- median(dat$year, na.rm=TRUE)

#categorical variable: odometer
dat$odometer <- cut(dat$odometer, breaks=c(0, 50000, 100000, 150000, 200000, 250000,
                        300000, 350000, 400000, 500000), labels=c(0, 1, 2, 3, 4, 5, 6, 7, 8))
mapping <- c("0" = 0, "1" = 1, "2" = 2, "3" = 3, "4" = 4, "5" = 5, "6" = 6, "7" = 7, "8" = 8)
```

```
dat$odometer <- mapping[dat$odometer]
dat$odometer[is.na(dat$odometer)] <- 9

#categorical variable: cylinders
unique(dat$cylinders)
```

```
## [1]  8 NA  4  6 10  3  5 12
```

```
dat$cylinders[is.na(dat$cylinders)] <- 9

#categorical variable: drive
unique(dat$drive)
```

```
## [1] "4wd" ""    "fwd" "rwd"
```

```
mapping <- c("fwd" = 0, "rwd" = 1, "4wd" = 2)
dat$drive <- mapping[dat$drive]
dat$drive[is.na(dat$drive)] <- 9

#categorical variable: manufacturer
dat$manufacturer <- as.factor(dat$manufacturer)
n <- length(unique(dat$manufacturer))
mapping <- c(0:n)
dat$manufacturer <- mapping[dat$manufacturer]
unique(dat$manufacturer)
```

```
##  [1] 13 38 23 20 30  7 39 11  8 19 16 24  9 14 31 29 33  0 26  6 17 28 22 21  5
## [26] 36 25 18 34 12 37  1 40 27 35  4 32  3  2 15 10
```

```
# generate an add-in variable: goodcredit
dat$description_goodcredit <- 0
dat$description_goodcredit[dat$description_credit == 1 & dat$description_badcredit == 0] <- 1
```

Step 2. Bagging, which is a special case of a random forest where m = p

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(1)

# Bagging
model<-lprice~year+odometer+transmission+condition+fuel+cylinders+drive+manufacturer+description_badcre
bag.boston = randomForest(model, data = dat, mtry = 7,  importance =TRUE)
print(bag.boston)
```

```
##
## Call:
##  randomForest(formula = model, data = dat, mtry = 7, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##           Mean of squared residuals: 0.1326835
##                     % Var explained: 77.82
```

Step 3. Generate 30,000 predictions

```
testdat <- read.csv("C:/Users/linzh/Desktop/ECON 626/hw3/final_predcomp_training_data_large.csv")
testdat$lprice <- log(testdat$price)

#categorical variable: fuel
unique(testdat$fuel)
```

```
## [1] "diesel"   "gas"      "hybrid"   "other"    ""         "electric"
```

```
mapping <- c("gas" = 0, "diesel" = 1, "hybrid" = 2, "electric" = 3, "other" = 4)
testdat$fuel <- mapping[testdat$fuel]
testdat$fuel[is.na(testdat$fuel)] <- 9

#categorical variable: transmission
unique(testdat$transmission)
```

```
## [1] "automatic" "other"     "manual"    ""
```

```
mapping <- c("automatic" = 0, "manual" = 1, "other" = 2)
testdat$transmission <- mapping[testdat$transmission]
testdat$transmission[is.na(testdat$transmission)] <- 9

#categorical variable: condition
unique(testdat$condition)
```

```
## [1] ""          "good"      "excellent" "like new"  "fair"      "new"
## [7] "salvage"
```

```
mapping <- c("new" = 0, "excellent" = 1, "good" = 2, "like new" = 3, "fair" = 4, "salvage" = 5)
testdat$condition <- mapping[testdat$condition]
testdat$condition[is.na(testdat$condition)] <- 9

# fill in na in year
testdat$year[is.na(testdat$year)] <- median(testdat$year, na.rm=TRUE)

# save odometer for Q3
odometer <- data.frame(testdat$lprice, testdat$odometer)

#categorical variable: odometer
testdat$odometer <- cut(testdat$odometer, breaks=c(0, 50000, 100000, 150000, 200000, 250000,
```

```
                            300000, 350000, 400000, 500000), labels=c(0, 1, 2, 3, 4, 5, 6, 7, 8))
mapping <- c("0" = 0, "1" = 1, "2" = 2, "3" = 3, "4" = 4, "5" = 5, "6" = 6, "7" = 7, "8" = 8)
testdat$odometer <- mapping[testdat$odometer]
testdat$odometer[is.na(testdat$odometer)] <- 9

#categorical variable: cylinders
unique(testdat$cylinders)
```

```
## [1]  8 NA  4  6 10  3  5 12
```

```
testdat$cylinders[is.na(testdat$cylinders)] <- 9

#categorical variable: drive
unique(testdat$drive)
```

```
## [1] "4wd" ""    "fwd" "rwd"
```

```
mapping <- c("fwd" = 0, "rwd" = 1, "4wd" = 2)
testdat$drive <- mapping[testdat$drive]
testdat$drive[is.na(testdat$drive)] <- 9

#categorical variable: manufacturer
testdat$manufacturer <- as.factor(testdat$manufacturer)
n <- length(unique(testdat$manufacturer))
mapping <- c(0:n)
testdat$manufacturer <- mapping[testdat$manufacturer]
unique(testdat$manufacturer)
```

```
##  [1] 14 40 25 22 32  7 41 11  8 21 18 26  9 15 33 31 35  0 28  6 19 30 24 23  5
## [26] 38 27 20 36 13 39  1 42 29 37  4 34  3  2 16 10 12 17
```

```
# generate an add-in variable: goodcredit
testdat$description_goodcredit <- 0
testdat$description_goodcredit[testdat$description_credit==1 & testdat$description_badcredit==0]<-1

# 30,000 predictions
bagged_estimate <- predict(bag.boston, newdata = testdat)

mean((bagged_estimate - testdat$lprice)^2) # MSE 0.1410525
```

```
## [1] 0.1410525
```

```
SSE <- sum((bagged_estimate - testdat$lprice)^2)
SST <- sum((testdat$lprice - mean(testdat$lprice))^2)
R_square <- 1 - SSE / SST #R_square
R_square # 0.7631135
```

```
## [1] 0.7631135
```

Q2) Draw a figure that demonstrates which features are most important in constructing more accurate predictions.

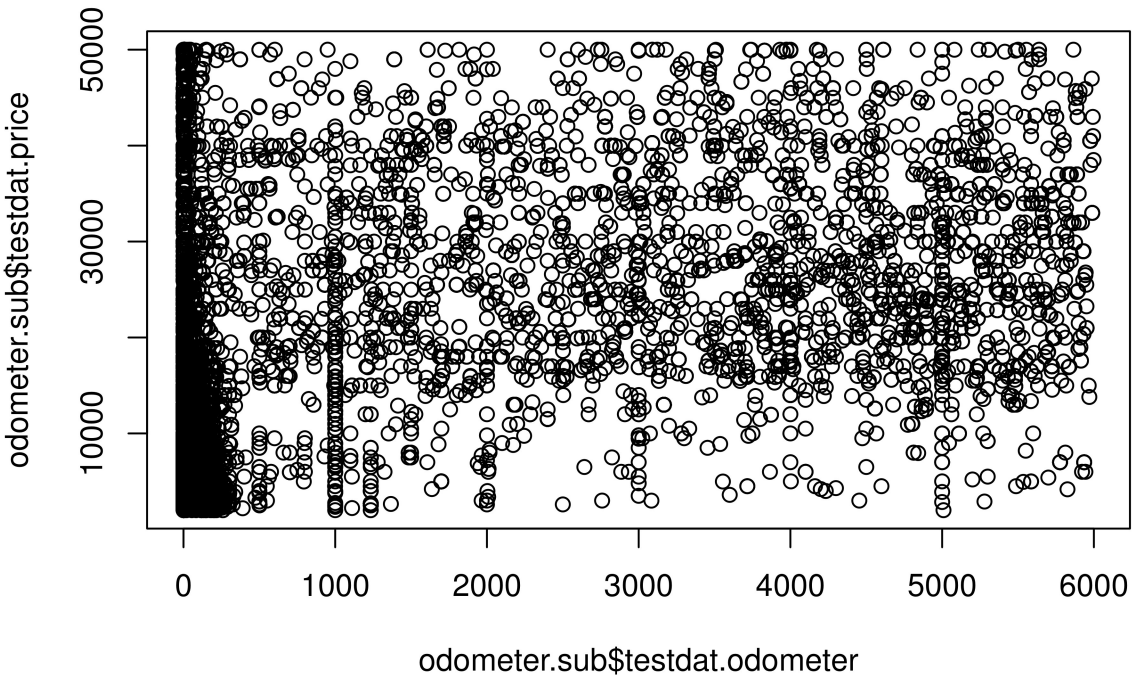
```
importance(bag.boston)
```

```
##                           %IncMSE IncNodePurity
## year                    536.46714     7891.6295
## odometer                227.42212     2476.1612
## transmission            112.90937      260.9559
## condition               151.51402      537.2442
## fuel                    290.75585      915.2276
## cylinders               256.19795     1094.5832
## drive                   308.28558     1799.2319
## manufacturer            249.41587     1434.2621
## description_badcredit    90.20778      175.5652
## description_goodcredit   83.36587      311.7649
```

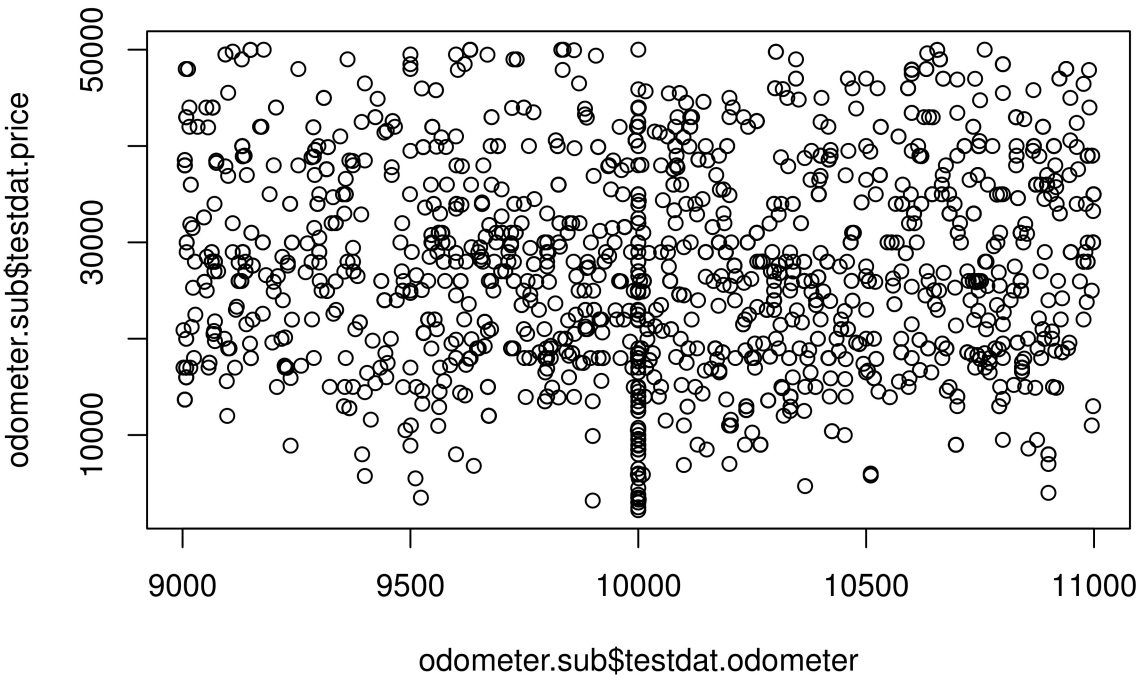
```
varImpPlot(bag.boston)
```

Q3

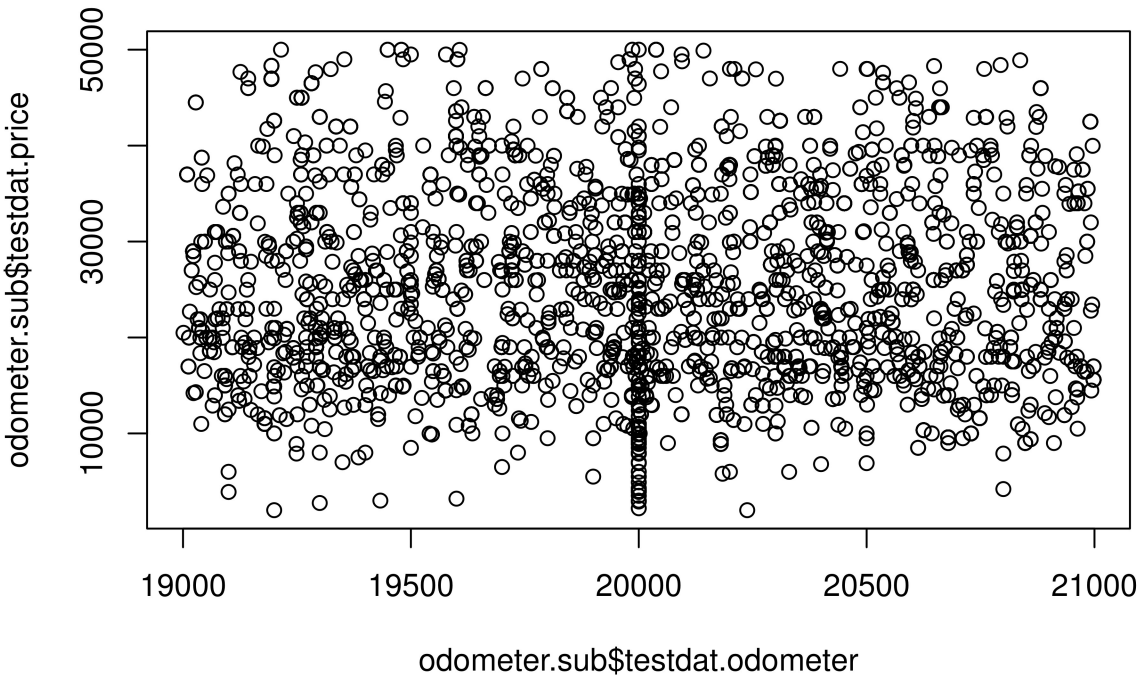
```
odometer <- data.frame(odometer, testdat$price)

odometer.sub <- subset(odometer, odometer$testdat.odometer < 6000)
plot(odometer.sub$testdat.odometer, odometer.sub$testdat.price)
```



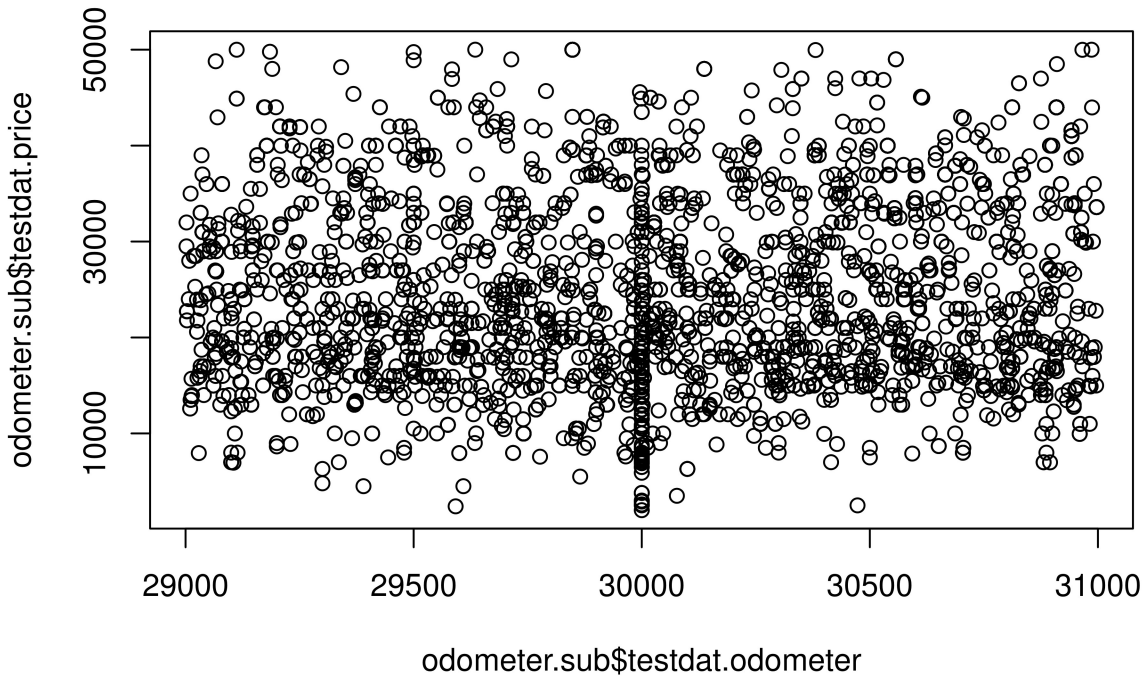

```
odometer.sub <- subset(odometer, odometer$testdat.odometer < 11000 & odometer$testdat.odometer > 9000)
plot(odometer.sub$testdat.odometer, odometer.sub$testdat.price)
```



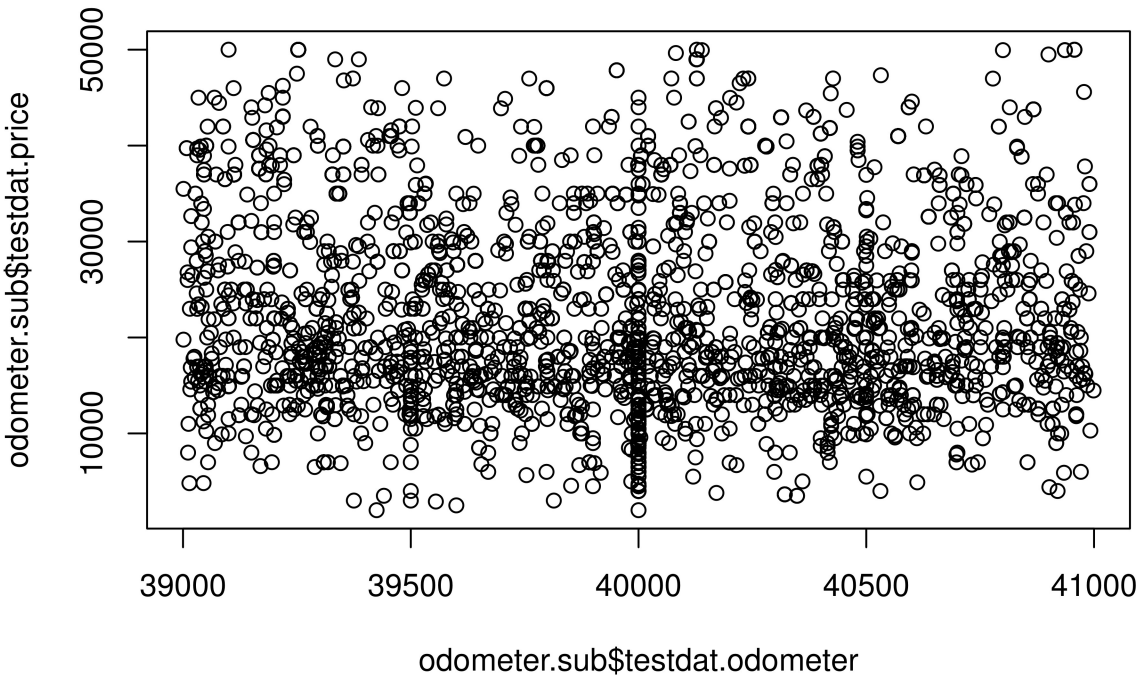

```
odometer.sub <- subset(odometer, odometer$testdat.odometer < 21000 & odometer$testdat.odometer > 19000)
plot(odometer.sub$testdat.odometer, odometer.sub$testdat.price)
```




```
odometer.sub <- subset(odometer, odometer$testdat.odometer < 31000 & odometer$testdat.odometer > 29000)
plot(odometer.sub$testdat.odometer, odometer.sub$testdat.price)
```




```
odometer.sub <- subset(odometer, odometer$testdat.odometer < 41000 & odometer$testdat.odometer > 39000)
plot(odometer.sub$testdat.odometer, odometer.sub$testdat.price)
```



In the article, the author pointed out the partial inattention to mileage has a significant effect on the used-car prices. From the result, I think this pattern holds for the Craigslist data. We can clearly see that at odometer = 1000, 5000, 10000, 20000, 3000, 40000, the partial inattention to odometer has a significant effect on prices.