assignment1

yunzhi wang

2/27/2018

1.import data

```
library(caret)

## Warning: package 'caret' was built under R version 3.4.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2018c
.
## 1.0/zoneinfo/America/Chicago'

data("GermanCredit")
mydata <- GermanCredit
```

2.perform regression model

```
y <- "Amount"
available.x <- colnames(mydata)[-2]
optimal.x <- NULL
r2 <- NULL


while (length(available.x) > 0) {
  best.r2 <- 0
  for (this.x in available.x) {
    rhs <- paste(c(optimal.x, this.x), collapse=" + ")
    f <- as.formula(paste(y, rhs, sep=" ~ "))
    this.r2 <- summary(lm(f, data=mydata))$r.square
    if (this.r2 > best.r2) {
      best.r2 <- this.r2
      best.x <- this.x
    }
  }
  optimal.x <- c(optimal.x, best.x)
  available.x <- available.x[available.x != best.x]
  r2 <- c(r2, best.r2)
}

optimal.x <- c("(Intercept)", optimal.x)
r2 <- c(summary(lm(Amount ~ 1, data=mydata))$r.square, r2)
```

```r
cum.r2 <- cbind(optimal.x, r2)
#cum.r2

#I chose the top 6 elements (intercept included) with culmulative r squared o
f 57%.
mypredictor <- cum.r2[1:6, 1]
mypredictor
```

```
## [1] "(Intercept)"
## [2] "Duration"
## [3] "InstallmentRatePercentage"
## [4] "Job.Management.SelfEmp.HighlyQualified"
## [5] "Personal.Male.Single"
## [6] "Telephone"
```

3. save all the results

```r
set.seed(711)
mydata.split <- replicate(1000, sample(1:nrow(mydata), size = 0.632 * nrow(my
data)))

head(mydata.split[,1])
```

```
## [1] 290 427 325 629 444 123
```

```r
#Create a data frame that stores result
result <- data.frame(matrix(ncol = 9, nrow = 1000))
colnames(result) <- c("Intercept", "Duration","InstallmentRatePercentage",
                      "Job.Management.SelfEmp.HighlyQualified",
                      "Personal.Male.Single","Telephone", "r.training",
                      "r.testing", "percent.r.fall")
```

4. Make a For loop:split training and testing samples, apply linear model, get model coefficients, r-squared training and r-squared.testing and save all of these.

```r
for (i in 1:1000) {
  #split
  training <- mydata[mydata.split[,i], c(1,2,3,8,43,62)]
  testing <- mydata[-mydata.split[,i], c(1,2,3,8,43,62)]
  #linear model
  linearmodel <- lm(Amount ~ Duration + InstallmentRatePercentage +
                      Job.Management.SelfEmp.HighlyQualified +
                      Personal.Male.Single +
                      Telephone, data = training)
  #Coefficients
  coefficients <- linearmodel$coefficients
  #r-squared training
  r.squared.training <- summary(linearmodel)$r.squared
  #r-squared testing
  prediction <- predict(linearmodel, testing)
  sse <- sum((testing[, 2] - prediction) ^ 2)
  sst <- sum((testing[, 2] - mean(testing[,2])) ^ 2)
```

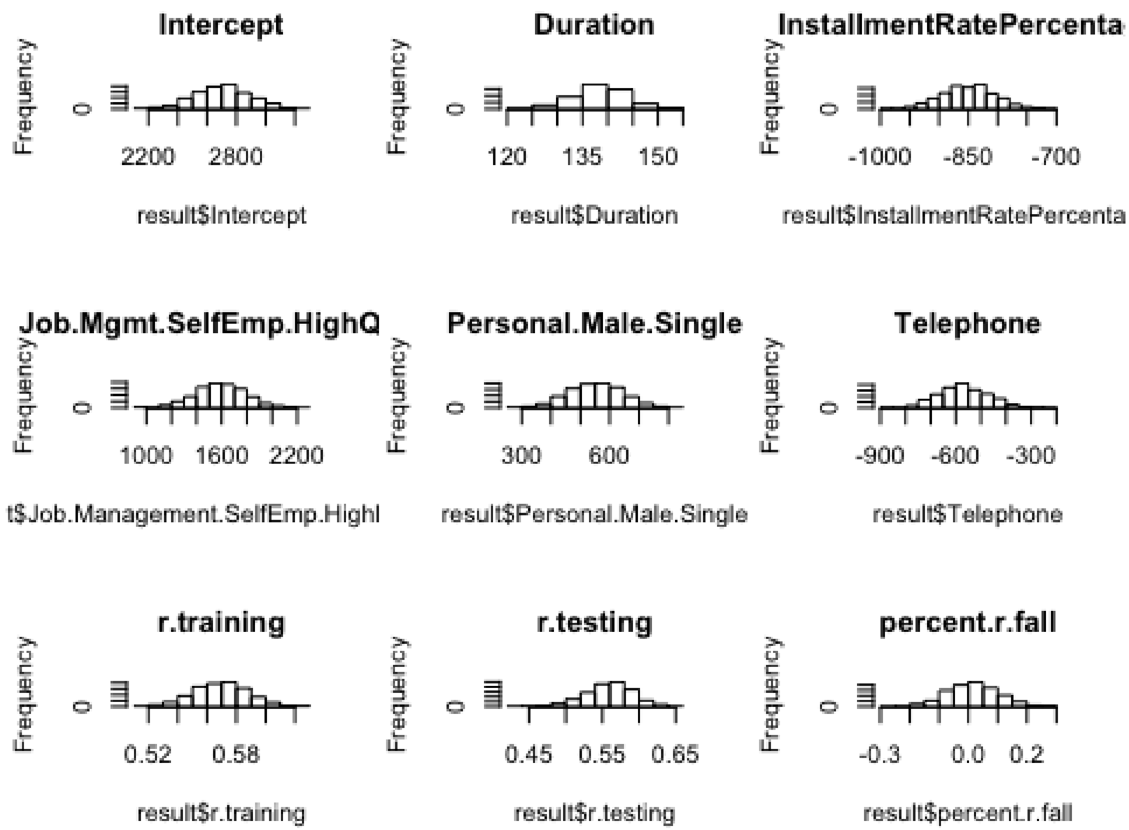
```
  r.squared.test <- 1 - (sse/sst)
  percent.r.fall <-(r.squared.training-r.squared.test)/r.squared.training
  #Save data into dataframe result
  sample.c <- c(coefficients, r.squared.training, r.squared.test,
                percent.r.fall)
  result[i,] <- t(sample.c)
}

head(result)

##   Intercept Duration InstallmentRatePercentage
## 1  2583.775 141.5555                 -822.8759
## 2  2666.389 142.1681                 -851.8195
## 3  2443.578 130.5099                 -776.1379
## 4  2585.095 147.5275                 -840.0481
## 5  2802.072 137.2650                 -890.3304
## 6  2680.280 141.7962                 -822.2553
##   Job.Management.SelfEmp.HighlyQualified Personal.Male.Single Telephone
## 1                               1665.436             560.1621 -571.8271
## 2                               1344.809             690.5680 -658.4856
## 3                               1395.637             691.0189 -394.3792
## 4                               1525.020             616.3186 -720.3317
## 5                               1731.848             499.3024 -452.8767
## 6                               1559.377             406.0203 -627.7269
##   r.training r.testing percent.r.fall
## 1  0.5733258 0.5570834     0.02833009
## 2  0.5698756 0.5606138     0.01625229
## 3  0.5723469 0.5510505     0.03720893
## 4  0.5854411 0.5146136     0.12098144
## 5  0.5493247 0.6015989    -0.09516081
## 6  0.5703305 0.5592555     0.01941861
```

5.Plot distribution of all coefficients, holdou tr-squared and train r-squared

```
par(mfrow = c(3, 3))
hist(result$Intercept, main = "Intercept")
hist(result$Duration, main ="Duration")
hist(result$InstallmentRatePercentage, main ="InstallmentRatePercentage")
hist(result$Job.Management.SelfEmp.HighlyQualified,
     main ="Job.Mgmt.SelfEmp.HighQual")
hist(result$Personal.Male.Single, main ="Personal.Male.Single")
hist(result$Telephone, main ="Telephone")
hist(result$r.training, main ="r.training")
hist(result$r.testing, main ="r.testing")
hist(result$percent.r.fall, main ="percent.r.fall")
```



6. Compute average and standard deviation of each coefficient

```
coef.aveNSd <- data.frame(Mean = apply(result[,1:6], 2, mean),
                          Sd = apply(result[,1:6], 2, sd))
coef.aveNSd
```

```
##                                          Mean         Sd
## Intercept                           2706.5776 184.339136
## Duration                             138.4097   5.203162
## InstallmentRatePercentage           -850.0564  44.548587
## Job.Management.SelfEmp.HighlyQualified 1589.7627 201.054088
## Personal.Male.Single                 551.2444  91.462008
## Telephone                           -569.0529  99.701175
```

```
r.sq.aveNSd <- data.frame(Mean = apply(result[,7:9], 2, mean),
                           Sd = apply(result[,7:9], 2, sd))
r.sq.aveNSd
```

```
##                      Mean         Sd
## r.training     0.56905102 0.01819115
## r.testing      0.55869096 0.03229697
## percent.r.fall 0.01544784 0.08740334
```

7. compute average of 1000 to single model built using entire sample.

```r
#sample data
SampleData <- mydata[,c(1,2,3,8,43,62)]
#Apply linear model
linearmodel <- lm(Amount ~ Duration + InstallmentRatePercentage +
                     Job.Management.SelfEmp.HighlyQualified +
                     Personal.Male.Single +
                     Telephone, data = SampleData)
#Coefficients
coefficients.sampledata <- linearmodel$coefficients
#r-squared
r.squared.sampledata <- summary(linearmodel)$r.squared
entireSample.cur <- c(coefficients.sampledata, r.squared.sampledata = r.squar
ed.sampledata)
```

8. 95% confidence interval for coefficients.

```r
#Training/Testing Data
CI <- function(a) {
  lower <- coef.aveNSd$Mean[a] - qnorm(0.975)*coef.aveNSd$Sd[a]/sqrt(1000)
  upper <- coef.aveNSd$Mean[a] + qnorm(0.975)*coef.aveNSd$Sd[a]/sqrt(1000)
  c.i <- c(lower, upper)
  return(c.i)
}
ci.result.scaled <-data.frame(matrix(nrow = 6, ncol = 2))
colnames(ci.result.scaled) <- c("CI.lower.split", "CI.upper.split")
rownames(ci.result.scaled) <- c("Intercept","Duration",
                                "InstallmentRatePercentage",
                                "Job.Management.SelfEmp.HighlyQualified",
                                "Personal.Male.Single","Telephone")
ci.result.scaled[1:6,] <- rbind(CI(1), CI(2), CI(3), CI(4), CI(5), CI(6))
ci.result.scaled[,1] <- ci.result.scaled[,1]*(0.632^0.5)
ci.result.scaled[,2] <- ci.result.scaled[,2]*(0.632^0.5)
ci.result.scaled$range <- ci.result.scaled$CI.upper.split - ci.result.scaled$
CI.lower.split
ci.result.scaled

##                                        CI.lower.split CI.upper.split
## Intercept                                   2142.6038      2160.7696
## Duration                                     109.7772       110.2899
## InstallmentRatePercentage                   -677.9765      -673.5865
## Job.Management.SelfEmp.HighlyQualified      1253.9299      1273.7428
## Personal.Male.Single                         433.7240       442.7372
## Telephone                                   -457.3007      -447.4756
##                                            range
## Intercept                              18.165785
## Duration                                0.512748
## InstallmentRatePercentage               4.390061
## Job.Management.SelfEmp.HighlyQualified 19.812968
## Personal.Male.Single                    9.013166
## Telephone                               9.825098
```

```r
#Entire sample
ci.result.entire <-data.frame(matrix(nrow = 6, ncol = 2))
colnames(ci.result.entire) <- c("CI.lower.entire", "CI.upper.entire")
rownames(ci.result.entire) <- c("Intercept","Duration",
                                "InstallmentRatePercentage",
                                "Job.Management.SelfEmp.HighlyQualified",
                                "Personal.Male.Single","Telephone")
summary(linearmodel)

##
## Call:
## lm(formula = Amount ~ Duration + InstallmentRatePercentage +
##     Job.Management.SelfEmp.HighlyQualified + Personal.Male.Single +
##     Telephone, data = SampleData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4895.8 -1034.8  -175.7   676.6 11320.5
##
## Coefficients:
##                                        Estimate Std. Error t value
## (Intercept)                            2700.715    217.915  12.393
## Duration                                138.648      5.004  27.707
## InstallmentRatePercentage              -850.742     53.078 -16.028
## Job.Management.SelfEmp.HighlyQualified 1601.628    180.427   8.877
## Personal.Male.Single                    552.664    120.045   4.604
## Telephone                              -567.548    130.927  -4.335
##                                        Pr(>|t|)
## (Intercept)                             < 2e-16 ***
## Duration                                < 2e-16 ***
## InstallmentRatePercentage               < 2e-16 ***
## Job.Management.SelfEmp.HighlyQualified  < 2e-16 ***
## Personal.Male.Single                   4.69e-06 ***
## Telephone                              1.61e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1859 on 994 degrees of freedom
## Multiple R-squared:  0.5683, Adjusted R-squared:  0.5661
## F-statistic: 261.7 on 5 and 994 DF,  p-value: < 2.2e-16

ci.result.entire[1:6, ] <- c(coef(summary(linearmodel))[, 1] -
                               qnorm(0.975) *
                               coef(summary(linearmodel))[,2] / sqrt(1000),
                             coef(summary(linearmodel))[,1] + qnorm(0.975) *
                               coef(summary(linearmodel))[,2] / sqrt(1000))

ci.result.entire$range <- ci.result.entire$CI.upper.entire - ci.result.entire
$CI.lower.entire
ci.result.entire
```

```
##                                          CI.lower.entire CI.upper.entire
## Intercept                                     2687.2085       2714.2211
## Duration                                       138.3377        138.9580
## InstallmentRatePercentage                     -854.0315       -847.4521
## Job.Management.SelfEmp.HighlyQualified        1590.4457       1612.8113
## Personal.Male.Single                           545.2240        560.1046
## Telephone                                     -575.6628       -559.4332
##                                             range
## Intercept                              27.0125625
## Duration                                0.6202957
## InstallmentRatePercentage               6.5794842
## Job.Management.SelfEmp.HighlyQualified 22.3655209
## Personal.Male.Single                   14.8806012
## Telephone                              16.2295763
```

9. summary

10. I used the step wise method for the entire sample and chose the top 5 predictors who have a culmulative r^2 of 57%.

```
mypredictor <- cum.r2[2:6, 1]
```

2. The plots can show that each parameter follows central limit theorem and their distribution are mostly normal.

```
coef.aveNSd

##                                             Mean         Sd
## Intercept                              2706.5776 184.339136
## Duration                                138.4097   5.203162
## InstallmentRatePercentage              -850.0564  44.548587
## Job.Management.SelfEmp.HighlyQualified 1589.7627 201.054088
## Personal.Male.Single                    551.2444  91.462008
## Telephone                              -569.0529  99.701175

ci.result.scaled

##                                        CI.lower.split CI.upper.split
## Intercept                                   2142.6038      2160.7696
## Duration                                     109.7772       110.2899
## InstallmentRatePercentage                   -677.9765      -673.5865
## Job.Management.SelfEmp.HighlyQualified      1253.9299      1273.7428
## Personal.Male.Single                         433.7240       442.7372
## Telephone                                   -457.3007      -447.4756
##                                            range
## Intercept                              18.165785
## Duration                                0.512748
## InstallmentRatePercentage               4.390061
## Job.Management.SelfEmp.HighlyQualified 19.812968
## Personal.Male.Single                    9.013166
## Telephone                               9.825098
```

3. r.squared.testing has a wider distribution than r.squared.training, which shows that there is a higher variance in testing dataset,but the difference is not that big.

```
r.sq.aveNSd

##                     Mean         Sd
## r.training     0.56905102 0.01819115
## r.testing      0.55869096 0.03229697
## percent.r.fall 0.01544784 0.08740334
```

4. From the confidence interval of splitted and entire sample, we can see that in accordance with the bootstrap method that reduces the variance, splitted datasets has a narrower confidence interval than the entire sample.