assignment5-part1

yunzhi wang

2/26/2018

1. import data

```
set.seed(600)
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'

library(rpart)
data("GermanCredit")
mydata <- GermanCredit[,1:7]
mydata.split <- sample(1:nrow(mydata), size = 0.7 * nrow(mydata))
Train <- mydata[mydata.split,]
Holdout <- mydata[-mydata.split,]
```

clustreg model

```
clustreg=function(dat,k,tries,sed,niter){

set.seed(sed)
dat=as.data.frame(dat)
rsq=rep(NA,niter)
res=list()
rsq.best=0
    for(l in 1:tries) {

    c = sample(1:k,nrow(dat),replace=TRUE)
    yhat=rep(NA,nrow(dat))
    for(i in 1:niter) {
        resid=pred=matrix(0,nrow(dat),k)
        for(j in 1:k){
            pred[,j]=predict(glm(dat[c==j,],family="gaussian"),newdata=dat)
            resid[,j] = (pred[,j]-dat[,1])^2
        }

    c = apply(resid,1,fun.index.rowmin)
    for(m in 1:nrow(dat)) {yhat[m]=pred[m,c[m]]}
```

```
    rsq[i] = cor(dat[,1],yhat)^2
    #print(rsq[i])
    }

    if(rsq[niter] > rsq.best) {
        rsq.best=rsq[niter]
        l.best=l
            c.best=c
        yhat.best=yhat
        }
    }

    for(i in k:1) res[[i]]=summary(lm(dat[c.best==i,]))

return(list(data=dat,nclust=k,tries=tries,seed=sed,rsq.best=rsq.best,number.l
oops=niter, Best.try=l.best,cluster=c.best,results=res))
}
fun.index.rowmin=function(x) {

    z=(1:length(x)) [x == min(x)]
    if(length(z) > 1) { z=sample(z,1)}
    return ( z ) }
```

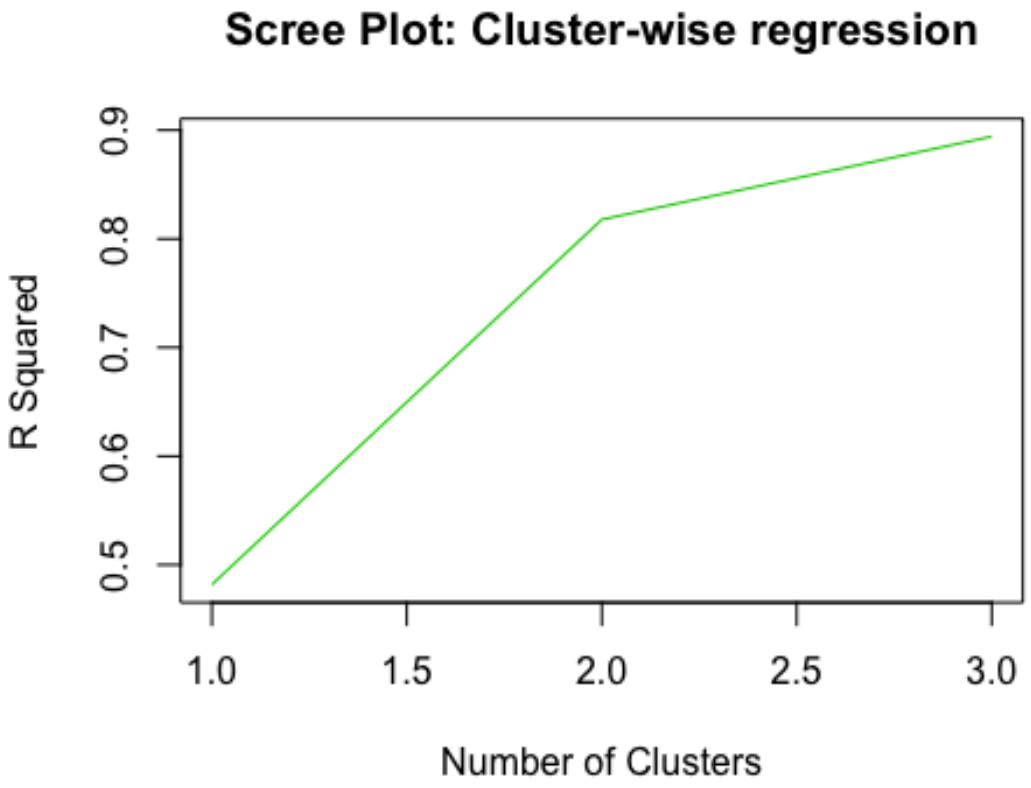
2.use Train data set to build clusterwise model， use clustreg(), use numeric variables, plot R-squared as a function of the clusters

```
set.seed(123)
cluster1 <- clustreg(Train, 1, 10, 711, 15)
cluster2 <- clustreg(Train, 2, 10, 711, 15)
cluster3 <- clustreg(Train, 3, 10, 711, 15)

rsq.cluster1 <- cluster1$rsq.best
rsq.cluster2 <- cluster2$rsq.best
rsq.cluster3 <- cluster3$rsq.best
rsq <- c(rsq.cluster1, rsq.cluster2, rsq.cluster3)
rsq

## [1] 0.4818167 0.8177472 0.8941116

par(mfrow = c(1, 1))
plot(1 : 3, rsq, main = "Scree Plot: Cluster-wise regression",
     xlab = "Number of Clusters", ylab = "R Squared", type = "l", col = "11")
```



3.

Perform holdout validation using clustreg.predict()

```
#clustreg.predict()
clustreg.predict=function(results,newdat){

    yhat=rep(NA,nrow(newdat))
    resid=pred=matrix(0,nrow(newdat),length(table(results$cluster)))

        for(j in 1:length(table(results$cluster))){

pred[,j]=predict(glm(results$data[results$cluster==j,],family="gaussian"),new
data=newdat)
            resid[,j] = (pred[,j]-newdat[,1])^2
        }

    c = apply(resid,1,fun.index.rowmin)
    for(m in 1:nrow(newdat)) {yhat[m]=pred[m,c[m]]}
    rsq = cor(newdat[,1],yhat)^2

return(list(results=results,newdata=newdat,cluster=c,yhat=yhat,rsq=rsq))

}
```

```
set.seed(123)
holdout.cluster1 <- clustreg.predict(cluster1, Holdout)
holdout.cluster2 <- clustreg.predict(cluster2, Holdout)
holdout.cluster3 <- clustreg.predict(cluster3, Holdout)

rsq.test <- c(holdout.cluster1$rsq, holdout.cluster2$rsq,
              holdout.cluster3$rsq)
rsq.test

## [1] 0.4050481 0.7839949 0.8795457
```

4. choose the model with the best performance in r-squared, training data and relevant significance

```
cluster1$results

## [[1]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -42.674  -5.518  -1.339   4.606  45.510
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.6991951  1.9567449   3.935 9.17e-05 ***
## Amount                     0.0030188  0.0001207  25.002  < 2e-16 ***
## InstallmentRatePercentage  2.6808223  0.3118035   8.598  < 2e-16 ***
## ResidenceDuration          0.1352130  0.3158979   0.428  0.66876
## Age                       -0.0791568  0.0305970  -2.587  0.00988 **
## NumberExistingCredits     -1.0901063  0.5773077  -1.888  0.05941 .
## NumberPeopleMaintenance   -0.7417621  0.9248208  -0.802  0.42279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.75 on 693 degrees of freedom
## Multiple R-squared:  0.4818, Adjusted R-squared:  0.4773
## F-statistic: 107.4 on 6 and 693 DF,  p-value: < 2.2e-16

#holdout.cluster1$results

cluster2$results

## [[1]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -11.495  -4.280  -1.733   2.997  34.813
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               14.6959379  2.3246330   6.322 1.24e-09 ***
## Amount                     0.0035020  0.0001536  22.806  < 2e-16 ***
## InstallmentRatePercentage  2.4506364  0.3931704   6.233 2.03e-09 ***
## ResidenceDuration          0.1289141  0.3800147   0.339   0.7347
## Age                       -0.0404566  0.0377240  -1.072   0.2846
## NumberExistingCredits     -4.3491349  0.6766535  -6.427 6.89e-10 ***
## NumberPeopleMaintenance    3.0581447  1.2303523   2.486   0.0136 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.272 on 241 degrees of freedom
## Multiple R-squared:  0.7064, Adjusted R-squared:  0.6991
## F-statistic: 96.65 on 6 and 241 DF,  p-value: < 2.2e-16
##
##
## [[2]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.5730  -2.6034  -0.0413   2.9910  10.8932
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.434e+00  1.286e+00   4.225 2.91e-05 ***
## Amount                    2.282e-03  7.697e-05  29.651  < 2e-16 ***
## InstallmentRatePercentage 2.096e+00  2.000e-01  10.480  < 2e-16 ***
## ResidenceDuration         3.463e-01  2.055e-01   1.685   0.0927 .
## Age                      -9.477e-02  1.973e-02  -4.804 2.13e-06 ***
## NumberExistingCredits    -6.248e-01  3.846e-01  -1.625   0.1050
## NumberPeopleMaintenance  -2.663e-01  5.784e-01  -0.460   0.6454
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.542 on 445 degrees of freedom
## Multiple R-squared:  0.6735, Adjusted R-squared:  0.6691
## F-statistic:   153 on 6 and 445 DF,  p-value: < 2.2e-16

#holdout.cluster2$results

cluster3$results
```

```
## [[1]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.866 -3.518 -0.019  2.082 32.500
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.110e+01  2.365e+00  13.153  < 2e-16 ***
## Amount                     3.642e-03  1.795e-04  20.290  < 2e-16 ***
## InstallmentRatePercentage  3.685e+00  4.175e-01   8.826 1.93e-15 ***
## ResidenceDuration          6.964e-01  4.144e-01   1.681   0.0948 .
## Age                       -9.800e-02  4.606e-02  -2.128   0.0349 *
## NumberExistingCredits      2.860e-01  7.891e-01   0.362   0.7175
## NumberPeopleMaintenance   -1.868e+01  9.844e-01 -18.972  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.442 on 158 degrees of freedom
## Multiple R-squared:  0.8806, Adjusted R-squared:  0.8761
## F-statistic: 194.3 on 6 and 158 DF,  p-value: < 2.2e-16
##
##
## [[2]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1034  -2.0541   0.3375   2.2066  13.0088
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.334e+00  1.389e+00   5.281 3.04e-07 ***
## Amount                     6.469e-04  8.729e-05   7.411 2.57e-12 ***
## InstallmentRatePercentage  3.913e+00  2.068e-01  18.920  < 2e-16 ***
## ResidenceDuration          5.724e-01  2.319e-01   2.468   0.0143 *
## Age                       -1.440e-01  2.030e-02  -7.094 1.71e-11 ***
## NumberExistingCredits      1.945e+00  4.388e-01   4.434 1.45e-05 ***
## NumberPeopleMaintenance    1.029e-01  6.931e-01   0.148   0.8821
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.53 on 223 degrees of freedom
## Multiple R-squared:  0.6775, Adjusted R-squared:  0.6688
## F-statistic: 78.08 on 6 and 223 DF,  p-value: < 2.2e-16
```

```
##
##
## [[3]]
##
## Call:
## lm(formula = dat[c.best == i, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.6177  -2.1372   0.2612   2.1363   9.4701
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1.937e+01  1.374e+00 -14.097  < 2e-16 ***
## Amount                     3.377e-03  6.793e-05  49.716  < 2e-16 ***
## InstallmentRatePercentage  2.560e+00  1.969e-01  13.001  < 2e-16 ***
## ResidenceDuration          8.783e-02  1.833e-01   0.479   0.6322
## Age                       -6.948e-02  1.749e-02  -3.972 8.94e-05 ***
## NumberExistingCredits      6.738e-01  3.235e-01   2.083   0.0381 *
## NumberPeopleMaintenance    1.858e+01  8.308e-01  22.359  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.364 on 298 degrees of freedom
## Multiple R-squared:  0.9154, Adjusted R-squared:  0.9137
## F-statistic: 537.7 on 6 and 298 DF,  p-value: < 2.2e-16

#holdout.cluster3$results
```

5.summarize the results for training and holdout

Interpretation: The model with one cluster, the train rsq is 48.2% and test rsq is 40.5%. This is not a strong result, it might be better if we consider the models which have more clusters.

For model of two clusters, it has train rsq of 81.8% and holdout rsq of 78.4%. The increase in the number of significant coefficients and good result of r-squared show that this might be a good result.

As for the three cluster model, it has a train rsq of 89.4% and test rsq with 88%. The rsq has a 7% improvement, meanwhile sacrifices one significant coefficient.

I would choose 3 cluster model based on its nice r-squared performance in train and test set.