BIOSTAT625 project test

Hengde Ouyang

2022-11-27

Read the Data and Remove the row contains NA

```
raw_data = read.csv("raw1.csv")
# Remove the row contains NA
complete_data = na.omit(raw_data)
```

Transform some of the data

```
transform_data = complete_data
# If the person is female, we code 0
transform_data$Sex = ifelse(transform_data$Sex=="Female",0,1)

# Survival months to numeric, and remove the unknown
# This is our response variable
transform_data$Survival.months = as.numeric(transform_data$Survival.months)
transform_data = na.omit(transform_data)

# Remove "years" in Single Age
# Code, if you want to remove:
# pattern = "years"
# transform_data$Age.recode.with.single.ages.and.85.  #=gsub(transform_data$Age.recode.with.single.ages
# replacement = "")

# Tumor Size to numeric, and remove the Blank
transform_data$CS.tumor.size..2004.2015. = as.numeric(transform_data$CS.tumor.size..2004.2015.)
transform_data = na.omit(transform_data)

# Tumor number to numeric, and remove the Blank
transform_data$Total.number.of.in.situ.malignant.tumors.for.patient = as.numeric(transform_data$Total.nu
transform_data = na.omit(transform_data)

### Silly Code
transform_data$Age.recode.with..1.year.olds = factor(transform_data$Age.recode.with..1.year.olds)
transform_data$Primary.Site...labeled = factor(transform_data$Primary.Site...labeled)
transform_data$Derived.AJCC.Stage.Group..6th.ed..2004.2015. = factor(transform_data$Derived.AJCC.Stage.
transform_data$ER.Status.Recode.Breast.Cancer..1990.. = factor(transform_data$ER.Status.Recode.Breast.Ca
transform_data$PR.Status.Recode.Breast.Cancer..1990.. = factor(transform_data$PR.Status.Recode.Breast.Ca
```

```r
transform_data$Survival.months.flag = factor(transform_data$Survival.months.flag)
transform_data$Race.ethnicity = factor(transform_data$Race.ethnicity)
```

Summary Statistics

```r
# summary(transform_data[,c(-6,-15)])
```

Training set and Testing set

```r
training = transform_data[,c(-6,-15)][transform_data$Year.of.diagnosis!=2015,]
testing = transform_data[,c(-6,-15)][transform_data$Year.of.diagnosis==2015,]
testing = testing[testing$Age.recode.with..1.year.olds!="05-09 years",]
```

MLR

```r
fit = lm(Survival.months~.,data = training)
```

```r
n = nrow(training)
RMSE_train = sqrt(sum((training$Survival.months - fit$fitted.values)^2)/n)
RMSE_train
```

```
## [1] 34.81291
```

```r
MLR_pred = predict(fit,testing)
```

```r
RMSE_test = sqrt(mean((testing$Survival.months-MLR_pred)^2))
RMSE_test
```

```
## [1] 20.28917
```

PCA

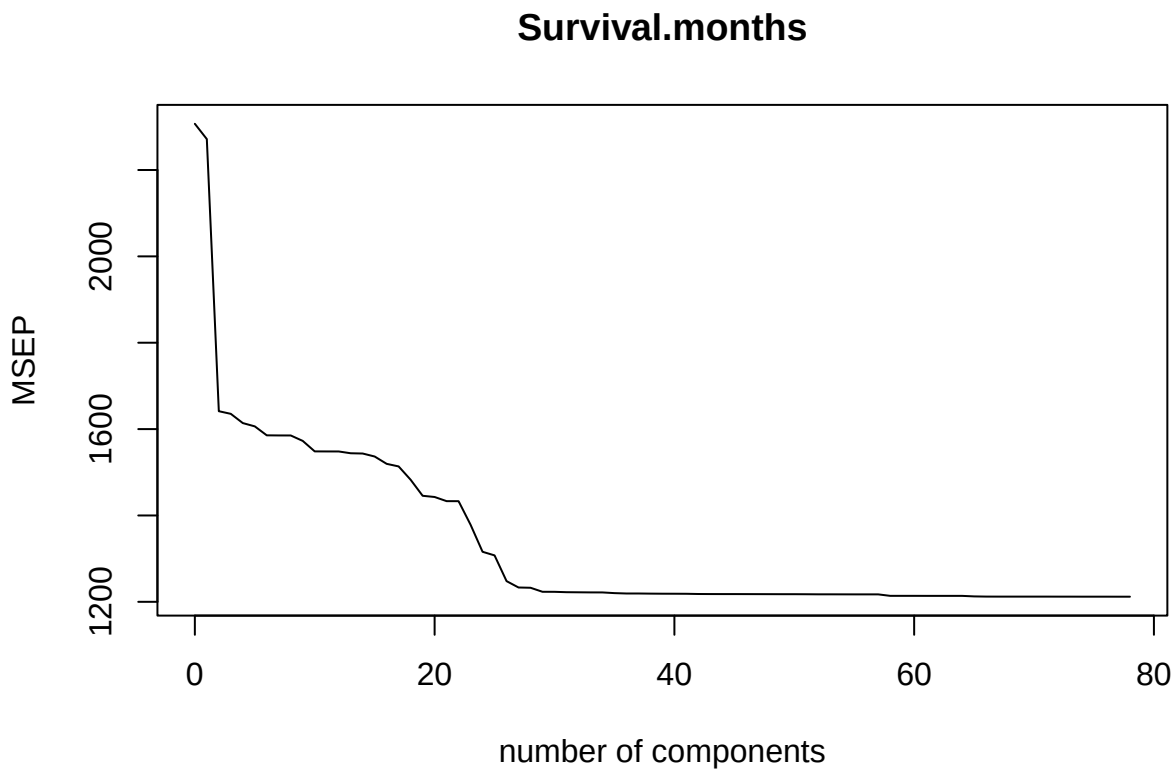
```r
library(pls)
```

```
##
##     'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
library(ISLR)
pcr.fit=pcr(Survival.months~.,data=training)
```

```
validationplot(pcr.fit ,val.type="MSEP")
```




```
pcr.fit=pcr(Survival.months~.,data=training,ncomp=10)
pcr.pred=predict (pcr.fit ,testing,ncomp =10)
RMSE2_test = sqrt(mean((testing$Survival.months-pcr.pred)^2))
```

Random Forest

```
#Before running the code, you need the "randomForest" package
#install.packages("randomForest")
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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