데이터분석과 R

Term Project

산업정보시스템공학과 201510336
구경민

제출일: 2020.12.18.Fri.

목차

1. Solar flare data set

1.1 solar flare
solar flare는 태양 흑점 근처에서 방출되는 복사 강도의 단기간동안 갑작스런 증가현상이다. solar flare는 수년 동안 H- 알파 파장에서 가장 잘 모니터링되었으며, 가끔 광구에서 백색광 플레어가 보이지만 색층에서 발생한다. 현대에서는 solar flare에 대해 위성을 통해 태양 X- 선 파장을 모니터링한다. flare는 몇 분 정도의 상승 시간과 수십 분 정도의 감쇠 시간이 특징이다. 일반적인 플레어에서 소비되는 총 에너지는 약 1030 에르그이며, 자기장은 100 ~ 10,000 가우스의 값에 도달하는 매우 높은 수준이다. H- 알파의 광학 플레어는 일반적으로 무선 및 X- 선 폭발을 동반하며 때로는 고 에너지 입자 방출을 동반한다.
flare의 광학 밝기와 크기는 "중요도"라는 두 문자 코드로 표시된다. 첫 번째 문자 인 1에서 4까지의 숫자는 명백한 영역을 나타낸다. 1보다 작은 영역의 경우 "S"가 하위 플레어를 지정하는 데 사용된다. 두 번째 문자는 상대적 밝기를 나타낸다. B는 밝음, N은 보통, F는 희미 함을 나타낸다. 이번 텀 프로젝트를 통해 흑점에 크기와 관련한 변수를 통해 흑점 크기를 분류하는 모델을 통해 흑점의 크기가 어떤 요인에 따라 변화되는지 알아보려고 한다.

1.2 Dataset 설명
flare 데이터는 2개의 데이터로 나뉘어져 있고(flare1, flare2) 여기서는 데이터변수를 flare1을 1969년 데이터라는 뜻으로 falre69 명명하고 1969년 데이터만 이용한다. 그리고 각 instance는 태양의 한번 활동한 영역 1개에 대한 feature다.

1.3 Attribute 설명
Class: Code for class (modified Zurich class), A,B,C,D,E,F,H로 구성
LSP: Code for largest spot size, X,R,S,A,H,K로 구성
SD: Code for spot distribution, X,O,I,C로 구성
Activity: 1 = reduced, 2 = unchanged
Evoluation: 1 = decay, 2 = no growth, 3 = growth
Activity24: Previous 24 hour flare activity code (1 = nothing as big as an M1, 2 = one M1, 3 = more activity than one M1)
HC: Historically-complex (1 = Yes, 2 = No)
RHC: Did region become historically complex (1 = yes, 2 = no) on this pass across the sun's disk
Area: 1 = small, 2 = large
ALS: Area of the largest spot (1 = <=5, 2 = >5)
아래는 추가변수들이다.
C_class: C-class flares production by this region in the following 24 hours (common flares); Number
M_class: M-class flares production by this region in the following 24 hours (moderate flares); Number
X_class: X-class flares production by this region in the following 24 hours (severe flares); Number

In [35]:

```
library(dplyr)
library(caret)
library(ggplot2)
```

dplyr은 전처리를 위한 도구, caret은 나이브베이지안 등 여러 분석 툴, ggplot2는 시각화를 위해 로드하였다.

In [36]:

```
library(readr)
flare69 <- read_table2("C:/Users/user/Desktop/비대면/데분알/MiddleTest/DataAnalysis_with_R/Term Project/fla
                       col_names = FALSE)
```

```
-- Column specification --------------------------------------------------------
cols(
  X1 = col_character(),
  X2 = col_character(),
  X3 = col_character(),
  X4 = col_double(),
  X5 = col_double(),
  X6 = col_double(),
  X7 = col_double(),
  X8 = col_double(),
  X9 = col_double(),
  X10 = col_double(),
  X11 = col_double(),
  X12 = col_double(),
  X13 = col_double()
)
```

2. Pre-processing & EDA

2.1 Pre-processing

먼저 데이터 컬럼에 이름을 부여하고, `head()`를 이용해서 데이터 상위 6개를 확인한다.

In [37]:

```
names(flare69) <- c('Class','LSP','SD','Activity',
                    'Evoluation','Activity24','HC',
                    'RHC','Area','ALS', 'C_class', 'M_class', 'X_class')
head(flare69)
```

A tibble: 6 × 13

Class	LSP	SD	Activity	Evoluation	Activity24	HC	RHC	Area	ALS	C_class	M_class	X_class
<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
C	S	O	1	2	1	1	2	1	2	0	0	0
D	S	O	1	3	1	1	2	1	2	0	0	0
C	S	O	1	3	1	1	2	1	1	0	0	0
D	S	O	1	3	1	1	2	1	2	0	0	0
D	A	O	1	3	1	1	2	1	2	0	0	0
D	A	O	1	2	1	1	2	1	2	0	0	0

`str()`을 이용해서 두 데이터의 구조를 확인한다.

In [38]:

```
print("flare69 summary")
summary(flare69)
print("flare69 str")
str(flare69)
```

```
[1] "flare69 summary"
    Class               LSP                SD              Activity
 Length:323         Length:323         Length:323         Min.   :1.000
 Class :character   Class :character   Class :character   1st Qu.:1.000
 Mode  :character   Mode  :character   Mode  :character   Median :1.000
                                                          Mean   :1.139
                                                          3rd Qu.:1.000
                                                          Max.   :2.000
   Evoluation      Activity24          HC             RHC
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
 1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000
 Median :3.000   Median :1.000   Median :1.000   Median :2.000
 Mean   :2.486   Mean   :1.192   Mean   :1.368   Mean   :1.947
 3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:2.000
 Max.   :3.000   Max.   :3.000   Max.   :2.000   Max.   :2.000
      Area            ALS           C_class          M_class
 Min.   :1.000   Min.   :1.000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:1.000   1st Qu.:2.000   1st Qu.:0.0000   1st Qu.:0.0000
 Median :1.000   Median :2.000   Median :0.0000   Median :0.0000
 Mean   :1.028   Mean   :1.755   Mean   :0.1331   Mean   :0.1362
 3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:0.0000   3rd Qu.:0.0000
 Max.   :2.000   Max.   :2.000   Max.   :2.0000   Max.   :4.0000
    X_class
 Min.   :0.00000
 1st Qu.:0.00000
 Median :0.00000
 Mean   :0.02167
 3rd Qu.:0.00000
 Max.   :1.00000
[1] "flare69 str"
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 323 obs. of  13 variables:
 $ Class     : chr  "C" "D" "C" "D" ...
 $ LSP       : chr  "S" "S" "S" "S" ...
 $ SD        : chr  "O" "O" "O" "O" ...
 $ Activity  : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Evoluation: num  2 3 3 3 3 2 2 2 3 3 ...
 $ Activity24: num  1 1 1 1 1 1 1 1 1 1 ...
 $ HC        : num  1 1 1 1 1 1 1 1 1 1 ...
 $ RHC       : num  2 2 2 2 2 2 2 2 2 2 ...
 $ Area      : num  1 1 1 1 1 1 1 1 1 1 ...
 $ ALS       : num  2 2 1 2 2 2 1 2 2 1 ...
 $ C_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ M_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ X_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "spec")=
  .. cols(
  ..   X1 = col_character(),
  ..   X2 = col_character(),
  ..   X3 = col_character(),
  ..   X4 = col_double(),
  ..   X5 = col_double(),
  ..   X6 = col_double(),
  ..   X7 = col_double(),
  ..   X8 = col_double(),
  ..   X9 = col_double(),
  ..   X10 = col_double(),
  ..   X11 = col_double(),
  ..   X12 = col_double(),
  ..   X13 = col_double()
  .. )
```

요인변수(factor)형이여야 하는 변수 class, LSP, SD가 char형이기에 as.factor를 이용해서 factor형으로 바꿔주고 다시 구조를 파악한다.

In [57]:

```
flare69$Class <- as.factor(flare69$Class)
flare69$LSP <- as.factor(flare69$LSP)
flare69$SD <- as.factor(flare69$SD)

str(flare69)
str(flare78)
```

```
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 323 obs. of  13 variables:
 $ Class     : Factor w/ 6 levels "B","C","D","E",..: 2 3 2 3 3 3 3 3 3 2 ...
 $ LSP       : Factor w/ 6 levels "A","H","K","R",..: 5 5 5 5 1 1 1 1 3 4 ...
 $ SD        : Factor w/ 4 levels "C","I","O","X": 3 3 3 3 3 3 3 3 3 3 ...
 $ Activity  : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Evoluation: num  2 3 3 3 3 2 2 2 3 3 ...
 $ Activity24: num  1 1 1 1 1 1 1 1 1 1 ...
 $ HC        : num  1 1 1 1 1 1 1 1 1 1 ...
 $ RHC       : num  2 2 2 2 2 2 2 2 2 2 ...
 $ Area      : num  1 1 1 1 1 1 1 1 1 1 ...
 $ ALS       : num  2 2 1 2 2 2 1 2 2 1 ...
 $ C_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ M_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ X_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "spec")=
  .. cols(
  ..   X1 = col_character(),
  ..   X2 = col_character(),
  ..   X3 = col_character(),
  ..   X4 = col_double(),
  ..   X5 = col_double(),
  ..   X6 = col_double(),
  ..   X7 = col_double(),
  ..   X8 = col_double(),
  ..   X9 = col_double(),
  ..   X10 = col_double(),
  ..   X11 = col_double(),
  ..   X12 = col_double(),
  ..   X13 = col_double()
  .. )
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 1066 obs. of  13 variables:
 $ Class     : Factor w/ 6 levels "B","C","D","E",..: 6 3 2 6 6 2 1 2 2 1 ...
 $ LSP       : Factor w/ 6 levels "A","H","K","R",..: 1 4 5 4 5 1 6 1 1 6 ...
 $ SD        : Factor w/ 4 levels "C","I","O","X": 4 3 3 4 4 3 3 3 3 3 ...
 $ Activity  : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Evoluation: num  3 3 3 2 1 2 3 3 2 3 ...
 $ Activity24: num  1 1 1 1 1 1 1 1 1 1 ...
 $ HC        : num  1 1 1 1 1 1 1 1 1 1 ...
 $ RHC       : num  1 2 2 1 2 2 2 2 2 2 ...
 $ Area      : num  1 1 1 1 1 1 1 1 1 1 ...
 $ ALS       : num  1 1 1 1 1 1 1 1 1 1 ...
 $ C_class   : num  0 0 0 0 0 0 0 0 1 0 ...
 $ M_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ X_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "spec")=
  .. cols(
  ..   X1 = col_character(),
  ..   X2 = col_character(),
  ..   X3 = col_character(),
  ..   X4 = col_double(),
  ..   X5 = col_double(),
  ..   X6 = col_double(),
  ..   X7 = col_double(),
  ..   X8 = col_double(),
  ..   X9 = col_double(),
  ..   X10 = col_double(),
  ..   X11 = col_double(),
  ..   X12 = col_double(),
  ..   X13 = col_double()
  .. )
```

데이터 구조를 확인하였으니 각 변수의 결측치와 이상치를 확인한다.

In [40]:

```
# 결측치 빈도확인
table(is.na(flare69))
table(is.na(flare78))
```

```
FALSE
 4199
FALSE
13858
```

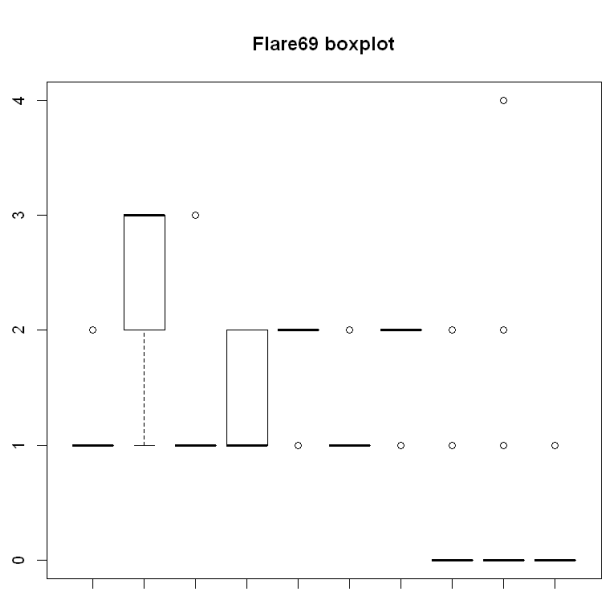
결측치가 없는 것을 확인할 수 있다.

2.2 EDA

2.1에서 전처리에서 데이터의 특징과 결측치의 유무를 확인하고 factor형으로 변환할 변수는 변환하였다. 다음은 상자그림을 통해 이상치를 파악하고, 모자이크 그림을 통해 빈도의 비례가 어떻게 되는지 파악한 다음에 상관분석을 통해 데이터간의 상관관계를 파악한다.

In [41]:

```
# 이상치 확인을 위한 boxplot
boxplot(flare69$Activity, flare69$Evoluation, flare69$Activity24,
        flare69$HC, flare69$RHC, flare69$Area, flare69$ALS, flare69$C_class,
        flare69$M_class, flare69$X_class,
       main='Flare69 boxplot')
```



각 변수에서 이상치가 나왔지만, 이상치의 데이터도 확인하면 4, 3, 2, 1 등 조사를 통해 낸 의미있는 데이터로 삭제하면 안된다. 결측치 제거는 넘어가는 것으로 했다.

그 다음으로 변수간의 상관관계를 알기 위한 상관분석에 앞서 flare69_1, flare78_1 변수에 각각 flare69와 flare78로 데이터를 복사해주고, factor인 변수를 numeric으로 변환한다. class의 A,B,C,D,E,F,H는 0,1,2,3,4,5,6으로 LSP의 X,R,S,A,H,K는 0,1,2,3,4,5로, SD의 X,O,I,C는 0,1,2,3으로 변환하여 상관분석을 행한다.

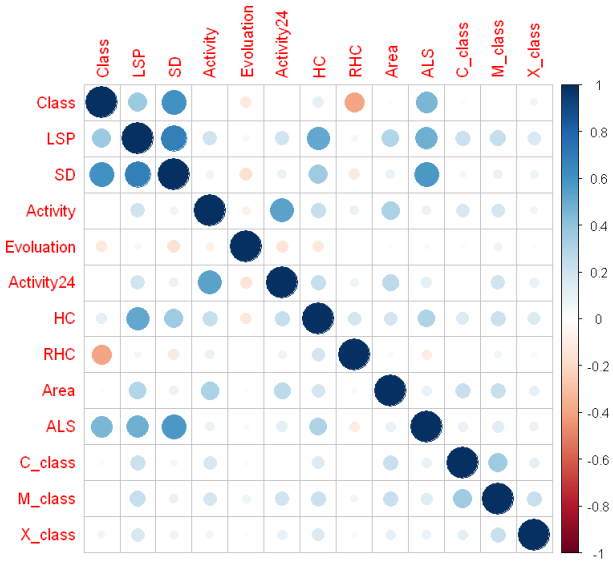
In [43]:

```
# correlation_preprocessing
flare69_1 <- flare69
flare69_1 <- within(flare69_1, {
            Class = ifelse(flare69_1$Class == 'A', 0,
                           ifelse(flare69_1$Class == 'B', 1,
                                  ifelse(flare69_1$Class == 'C', 2,
                                         ifelse(flare69_1$Class == 'D', 3,
                                                ifelse(flare69_1$Class == 'E', 4,
                                                       ifelse(flare69_1$Class == 'F',5, 6
                                                              ))))))
            LSP = ifelse(flare69_1$LSP == 'X', 0,
                         ifelse(flare69_1$LSP == 'R',1,
                                ifelse(flare69_1$LSP == 'S', 2,
                                       ifelse(flare69_1$LSP == 'A', 3,
                                              ifelse(flare69_1$LSP=='H',4, 5)))))
            SD = ifelse(flare69_1$LSP == 'X', 0,
                        ifelse(flare69_1$LSP == 'O', 1,
                               ifelse(flare69_1$LSP == 'I', 2, 3)))
})

str(flare69_1)
```

```
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 323 obs. of  13 variables:
 $ Class     : num  2 3 2 3 3 3 3 3 3 2 ...
 $ LSP       : num  2 2 2 2 3 3 3 3 5 1 ...
 $ SD        : num  3 3 3 3 3 3 3 3 3 3 ...
 $ Activity  : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Evoluation: num  2 3 3 3 3 2 2 2 3 3 ...
 $ Activity24: num  1 1 1 1 1 1 1 1 1 1 ...
 $ HC        : num  1 1 1 1 1 1 1 1 1 1 ...
 $ RHC       : num  2 2 2 2 2 2 2 2 2 2 ...
 $ Area      : num  1 1 1 1 1 1 1 1 1 1 ...
 $ ALS       : num  2 2 1 2 2 2 1 2 2 1 ...
 $ C_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ M_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ X_class   : num  0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "spec")=
  .. cols(
  ..   X1 = col_character(),
  ..   X2 = col_character(),
  ..   X3 = col_character(),
  ..   X4 = col_double(),
  ..   X5 = col_double(),
  ..   X6 = col_double(),
  ..   X7 = col_double(),
  ..   X8 = col_double(),
  ..   X9 = col_double(),
  ..   X10 = col_double(),
  ..   X11 = col_double(),
  ..   X12 = col_double(),
  ..   X13 = col_double()
  .. )
```

In [44]:

```
# correlation
library(corrplot)
flare69_cor <- cor(flare69_1)

corrplot(flare69_cor)
```



In [45]:

```
# correlation test (LSD and SD, HC, ALS)
cor.test(flare69_1$LSP, flare69_1$SD)
cor.test(flare69_1$LSP, flare69_1$HC)
cor.test(flare69_1$LSP, flare69_1$ALS)
```

```
 Pearson's product-moment correlation

data:  flare69_1$LSP and flare69_1$SD
t = 16.884, df = 321, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6233478 0.7395985
sample estimates:
      cor
0.6858235
 Pearson's product-moment correlation

data:  flare69_1$LSP and flare69_1$HC
t = 10.774, df = 321, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.430415 0.591218
sample estimates:
      cor
0.5153381
 Pearson's product-moment correlation

data:  flare69_1$LSP and flare69_1$ALS
t = 10.041, df = 321, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4011765 0.5677396
sample estimates:
      cor
0.4889014
```

LSP와 상관성이 있어 보이는 SD, HC, ALS 변수를 각각 상관분석을 행했을 때 유의수준보다 한참 낮은 값이 나왔기에 상관성이 있다고 알 수있다.

3. Modeling

3.1 흑점의 크기 분류를 위한 의사결정 나무

앞에서 상관분석을 한 결과 flare69 데이터에서 LSP(code for largest spot size)와, SD(code for spot distribution), HC(Historically-complex), ALS(Area of the largest spot)이 상관성을 보였다. 이에 이 변수를 이용하여 흑점의 크기를 분류하는 모델링 하여 insight를 도출한다. LSP는 이진이 아닌 명목변수로 되어있다. 즉, 종속변수 y가 이진변수여야 하는 logistic regression은 사용할 수 없다. 여기서는 명목변수가 이진이 아니어도 분류할 수 있는 의사결정 나무와 나이브베이지안 분류를 이용하여 모델링을 한 후, 성능을 비교하여 둘 중 가장 우수한 모델을 선택하여 insight를 도출한다.
의사결정 나무를 모델링하기 위해서 먼저 데이터를 학습데이터와 검증데이터를 분류해야한다. LSP와 상관성이 있는 데이터만 추출하고, 학습데이터, 검증데이터를 각 7:3으로 구분한 후에 모델링을 실시한다.

In [47]:

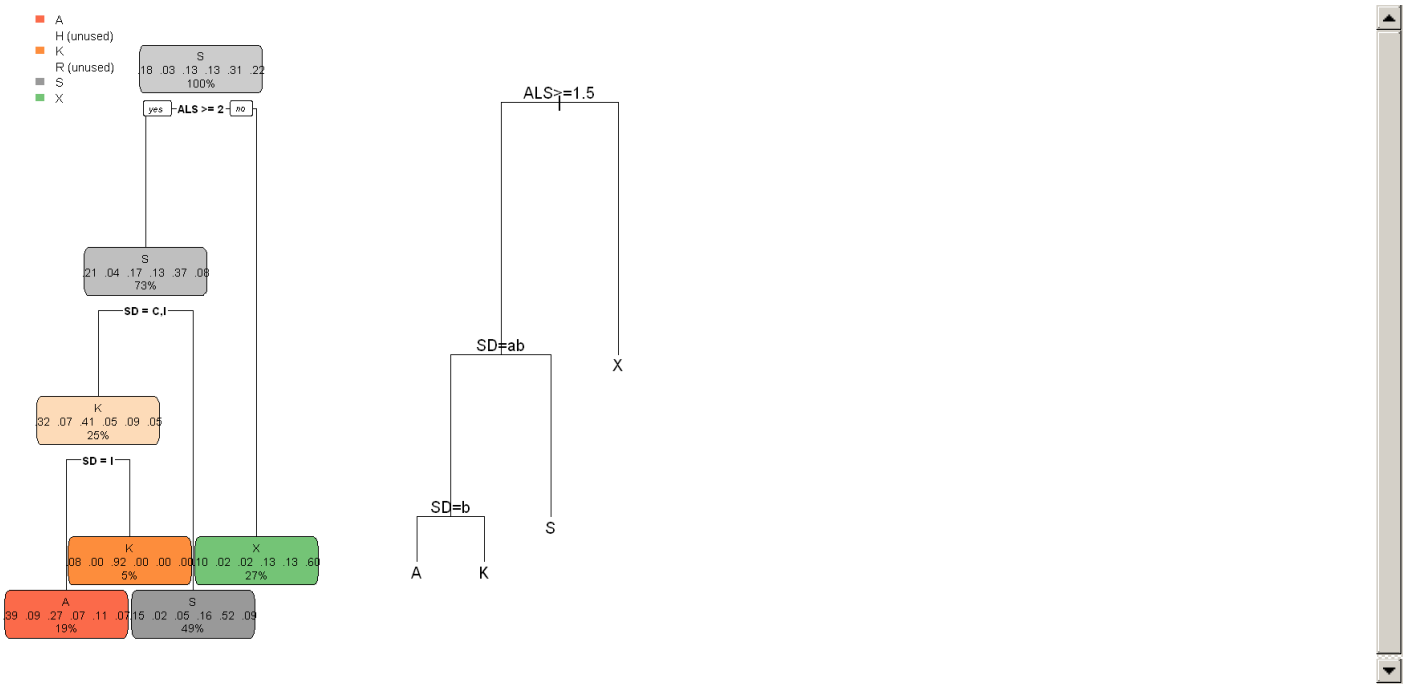
```
# separates data for spot size classification
# sc = Spot size Classification
sc_data <- flare69  %>%
    select(LSP, SD, HC, ALS)
set.seed(71)
# train data
sc_train <- sample_frac(sc_data, size=0.7)
# test data
sc_test <- flare69[setdiff(x=1:nrow(sc_data), y=sc_train),]
head(sc_train)
```

A tibble: 6 × 4

LSP	SD	HC	ALS
<fct>	<fct>	<dbl>	<dbl>
S	O	1	2
X	O	1	1
A	I	2	1
A	I	2	2
K	X	2	2
A	X	2	2

In [48]:

```
library(rpart)

# fitting
sct_fit <- rpart(LSP~., data=sc_train)
# plotting
library(rpart.plot) ; par(mfrow = c(1,2))
rpart.plot(sct_fit) ; plot(sct_fit) ; text(sct_fit)
```



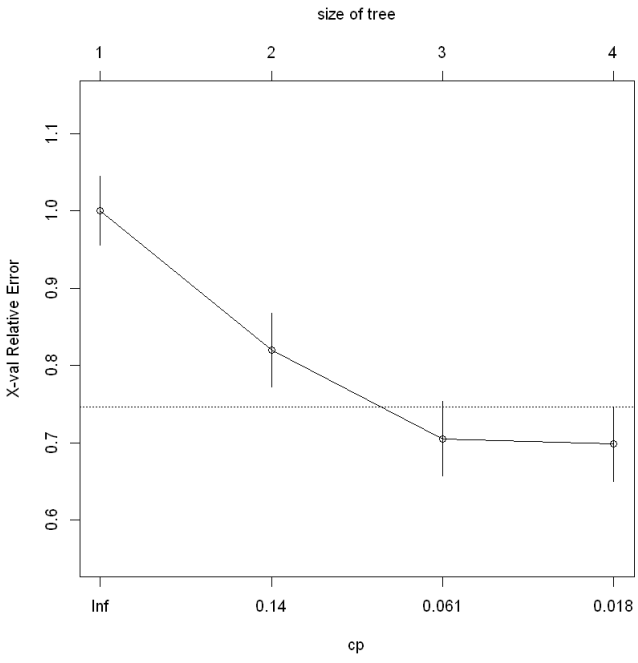
split 갯수가 3인 의사결정 나무가 나왔다. 최상위부터 ALS가 2 이상이 아니면 LSP가 X로 분류되고, 2이상이면 SD가 C,I 아니면 S로 C, I면 SD가 I면 A, 이외는 K로 분류되는 것을 볼 수 있다. 다음으로 테스트 데이터를 이용해서 예측률, 교차검증, 모델 정보를 알아보자

In [49]:

```
# testing
sct_pre <- predict(sct_fit, newdata =  sc_test, type='class')
sct_pre_testing <- sum(sct_pre==sc_test$LSP)/nrow(sc_test)*100
print("Validation model prediction")
sct_pre_testing

# 복잡성 확인
print("cptable")
print(sct_fit$cptable)

# 교차검증
print('Cross-validation')
printcp(sct_fit)
print(plotcp(sct_fit))

# 모델정보
print("Model information")
print(sct_fit$control)
```

```
[1] "Validation model prediction"
53.2507739938081
[1] "cptable"
          CP nsplit rel error    xerror       xstd
1 0.17948718      0 1.0000000 1.0000000 0.04455870
2 0.11538462      1 0.8205128 0.8205128 0.04775724
3 0.03205128      2 0.7051282 0.7051282 0.04816666
4 0.01000000      3 0.6730769 0.6987179 0.04815344
[1] "Cross-validation"

Classification tree:
rpart(formula = LSP ~ ., data = sc_train)

Variables actually used in tree construction:
[1] ALS SD

Root node error: 156/226 = 0.69027

n= 226
```

```
n= 226

        CP nsplit rel error  xerror     xstd
1 0.179487      0   1.00000 1.00000 0.044559
2 0.115385      1   0.82051 0.82051 0.047757
3 0.032051      2   0.70513 0.70513 0.048167
4 0.010000      3   0.67308 0.69872 0.048153
NULL
[1] "Model information"
$minsplit
[1] 20

$minbucket
[1] 7

$cp
[1] 0.01

$maxcompete
[1] 4

$maxsurrogate
[1] 5

$usesurrogate
[1] 2

$surrogatestyle
[1] 0

$maxdepth
[1] 30

$xval
[1] 10
```



prediction을 행했을 때 예측률이 53.25%로 비교적 예측률이 떨어지는 수치로 나왔다. 가지치기를 위해 행한 교차검증 결과에서 모델의 ERROR는 0.69로 error가 가장 낮은 split 갯수는 3으로 확인되었다. 하지만, 초기 fitting에서 split 갯수가 3인 모델이 나왔기에 prunning의 의미는 없다.

3.2 흑점의 크기 분류를 위한 나이브 베이지안 분류

앞에서 7:3으로 나눈 train data와 test data를 이용하여 나이브베이지안을 실시한다.

In [50]:

```
library(e1071)

# 나이브베이지안 피팅
scb_fit <- naiveBayes(LSP~., data=sc_train)
```

```
# 피팅 결과
summary(scb_fit)
```

```
          Length Class  Mode
apriori   6      table  numeric
tables    3      -none- list
levels    6      -none- character
isnumeric 3      -none- logical
call      4      -none- call
```

In [51]:

```
# prediction
scb_pre <- predict(scb_fit, newdata = sc_test)
scb_pre_testing <- sum(scb_pre==sc_test$LSP)/nrow(sc_test)*100
scb_pre_testing

# 정오분류표
scb_pre_table <- table(scb_pre, sc_data$LSP)
scb_pre_table
table(sc_data$LSP)

#정분류율
print("정분류율")
sum(diag(scb_pre_table))/sum(scb_pre_table)
```

54.4891640866873

```
scb_pre  A  H  K  R  S  X
      A 14  4  3  7  6  3
      H  0  0  0  0  0  0
      K 17  2 34  0  6  1
      R  0  0  0  1  0  0
      S 26  5 10 23 79 13
      X  4  0  1  7  9 48
  A   H   K   R   S   X 
 61  11  48  38 100  65 
[1] "정분류율"
```

0.544891640866873

나이브베이지안 결과 예측률이 54.489%로 의사결정 나무보다 약 1% 높은 예측률이 나왔다. 그리고 정분류율은 약 0.55로 나왔다.

4. Conclusion

흑점의 크기와 관련한 변수 LSP와 이와 상관성이 있는 변수들을 이용하여 의사결정 나무 그리고 나이브 베이지안 분류를 이용하여 모델링을 실시하였다. 두 모델 다 좋은 예측률이 나오지 않았다.
의사결정 나무는 흑점의 크기의 영역이 1 즉, 5 이하이면 LSP가 X로 분류되고, 5초과 되면 흑점의 분포 코드가 C,I 아니면 S로 C, I면 흑점의 분포코드가 I면 A, 이외는 K로 분류되었다. 그리고 나이브 베이지안으로 모델링한 결과 예측률은 54.49% 정분류율은 54.55%로 그렇게 높지 않은 분류예측이 나왔다.

출처
solar fire datasets
UCI Machine Learning Repository [Solar Flare Data set]
NOAA Solar features - Solar flares
위키백과 썬플레어
UCI Solar Flare data Set

위의 코드는 github에 게시하였습니다.
Ant9615/DataAnalysis_with_R/Term project