Classification.rmd

Classification

Goal1. Logistic Regression 모델 생성

Goal2. LDA 모델 생성

Goal3. SVM 모델 생성

Goal4. 각 모델의 비교

Data Loading and Setting

데이터 로딩

```
DATA <- read.table("KMOOC_2_04_dataset_iris.txt", header=TRUE)
```

데이터 탐색

```
dim(DATA)
```

```
## [1] 100   5
```

```
DATA[1:10,]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1      7.071079    3.539777     4.801605   1.8941238 versicolor
## 2      6.196075    3.875708     4.777633   1.0373698 versicolor
## 3      7.162724    3.156700     5.567441   2.7258140 versicolor
## 4      5.257869    1.538519     3.760564   0.2244236 versicolor
## 5      6.721558    3.223049     3.733300   1.6195305 versicolor
## 6      5.592241    2.804656     4.186294   0.7644110 versicolor
## 7      5.950493    3.015988     5.023600   0.6743164 versicolor
## 8      5.439054    2.651312     4.202546   1.1302752 versicolor
## 9      7.122297    3.637467     5.116050   1.4121607 versicolor
## 10     5.825488    2.635215     4.038487   1.4212949 versicolor
```

```
table(DATA$Species)
```

```
##
## versicolor  virginica
##         50         50
```

데이터 분리

```
train.idx <- c(1:30, 51:80)
data.train <- DATA[train.idx,]
data.test <- DATA[-train.idx,]
```

Logistic Regression

1개의 변수만 사용한 모델 생성 및 상관계수 확인

```
(f.slg <- glm(Species~Petal.Width, family=binomial(link='logit'), data=data.train))
```

```
##
## Call:  glm(formula = Species ~ Petal.Width, family = binomial(link = "logit"),
##     data = data.train)
##
## Coefficients:
## (Intercept)  Petal.Width
##      -3.449        2.062
##
## Degrees of Freedom: 59 Total (i.e. Null);  58 Residual
## Null Deviance:       83.18
## Residual Deviance: 64.61     AIC: 68.61
```

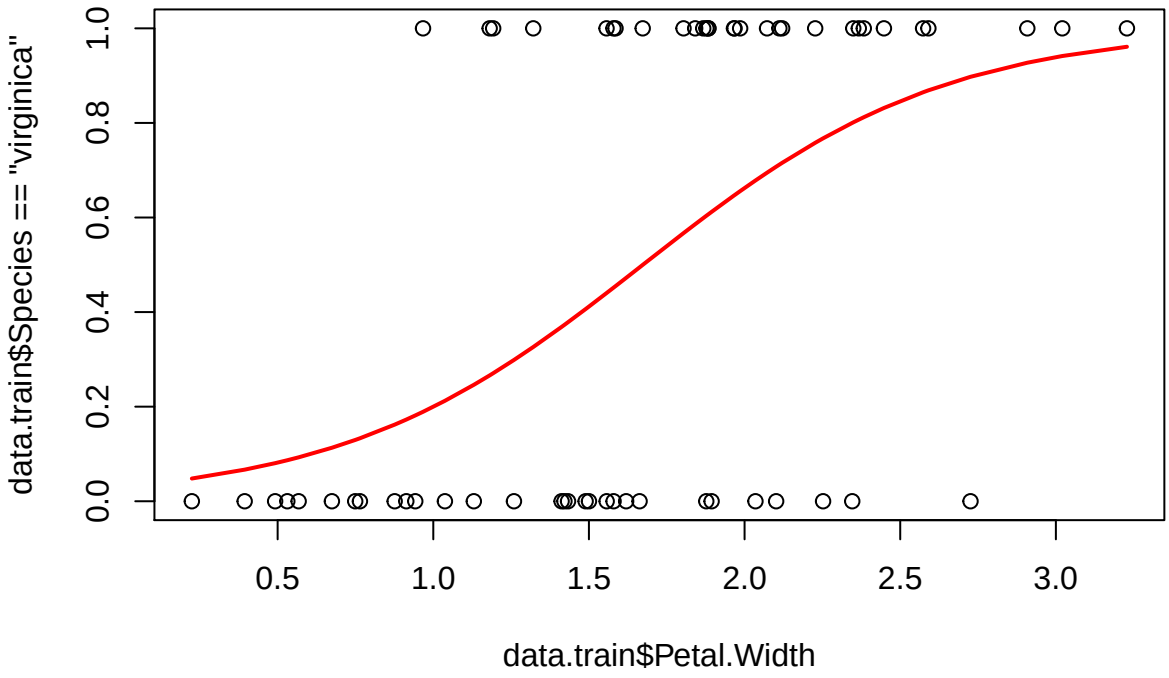
1개의 변수만 사용한 모델의 예측값 확인 (각 값은 virginica일 확률을 의미)

```
(y <- predict(f.slg, newdata=data.train, type='response'))
```

```
##          1          2          3          4          5          6          7
## 0.61195176 0.21236552 0.89753235 0.04803211 0.47238648 0.13314353 0.11312798
##          8          9         10         11         12         13         14
## 0.24616062 0.36863629 0.37302978 0.18136503 0.76740327 0.44042454 0.08053196
##         15         16         17         18         19         20         21
## 0.16200125 0.37862798 0.29871430 0.12956440 0.67827209 0.08672094 0.80008018
##         22         23         24         25         26         27         28
## 0.45172634 0.70742471 0.49460162 0.17271962 0.06680844 0.40641787 0.41175941
##         29         30         51         52         53         54         55
## 0.09285485 0.60351108 0.94145782 0.86888305 0.71558020 0.44019904 0.80693361
##         56         57         58         59         60         61         62
## 0.81204200 0.32614591 0.60704494 0.56697970 0.86481350 0.59867468 0.75794379
##         63         64         65         66         67         68         69
## 0.64737416 0.92732078 0.83155181 0.80116814 0.60374538 0.64646504 0.96106084
##         70         71         72         73         74         75         76
## 0.27069948 0.58575155 0.71182634 0.18913827 0.65576316 0.69478310 0.26611311
##         77         78         79         80
## 0.49980421 0.60643482 0.45490678 0.45149848
```

실제 값과 모델의 예측 확률 시각화 (virginica는 1로, virsicolor는 0으로 표현)

```
plot(data.train$Petal.Width, data.train$Species=='virginica')
o <- order(data.train$Petal.Width)
lines(data.train$Petal.Width[o], y[o], col=2, lwd=2)
```



확률에 따른 예측 인코딩 및 정확성 확인

```
(y.train.slg <- factor(y>0.5, levels=c(FALSE, TRUE), labels=c('versicolor', 'virginica')))
```

```
##          1          2          3          4          5          6          7
##  virginica versicolor  virginica versicolor versicolor versicolor versicolor
##          8          9         10         11         12         13         14
## versicolor versicolor versicolor versicolor  virginica versicolor versicolor
##         15         16         17         18         19         20         21
## versicolor versicolor versicolor versicolor  virginica versicolor  virginica
##         22         23         24         25         26         27         28
## versicolor  virginica versicolor versicolor versicolor versicolor versicolor
##         29         30         51         52         53         54         55
## versicolor  virginica  virginica  virginica  virginica versicolor  virginica
##         56         57         58         59         60         61         62
##  virginica versicolor  virginica  virginica  virginica  virginica  virginica
##         63         64         65         66         67         68         69
##  virginica  virginica  virginica  virginica  virginica  virginica  virginica
##         70         71         72         73         74         75         76
## versicolor  virginica  virginica versicolor  virginica  virginica versicolor
##         77         78         79         80
## versicolor  virginica versicolor versicolor
## Levels: versicolor virginica
```

```
table(y.train.slg, data.train$Species)
```

```
##
## y.train.slg  versicolor virginica
##   versicolor         23         8
##   virginica           7        22
```

test 세트에 대한 예측 및 정확성 확인

```
y <- predict(f.slg, newdata=data.test, type='response')
y.test.slg <- factor(y>0.5, levels=c(FALSE, TRUE), labels=c('versicolor', 'virginica'))
table(y.test.slg, data.test$Species)
```

```
##
## y.test.slg   versicolor virginica
##   versicolor         16         7
##   virginica           4        13
```

오차율 확인

```
(err.train.slg <- mean(y.train.slg != data.train$Species))
```

```
## [1] 0.25
```

```
(err.test.slg <- mean(y.test.slg != data.test$Species))
```

```
## [1] 0.275
```

모든 변수를 사용한 모델 생성 및 비교

```
f.lg <- glm(Species~., family=binomial(link='logit'), data=data.train)
y <- predict(f.lg, newdata=data.train, type='response')
y.train.lg <- factor(y>0.5, levels=c(FALSE, TRUE), labels=c('versicolor', 'virginica'))
y <- predict(f.lg, newdata=data.test, type='response')
y.test.lg <- factor(y>0.5, levels=c(FALSE, TRUE), labels=c('versicolor', 'virginica'))
table(y.train.lg, data.train$Species)
```

```
##
## y.train.lg   versicolor virginica
##   versicolor         26         5
##   virginica           4        25
```

```
table(y.test.lg, data.test$Species)
```

```
##
## y.test.lg    versicolor virginica
##   versicolor         17         9
##   virginica           3        11
```

```
(err.train.lg <- mean(y.train.lg != data.train$Species))
```

```
## [1] 0.15
```

```
(err.test.lg <- mean(y.test.lg != data.test$Species))
```

```
## [1] 0.3
```

LDA

패키지 로딩 및 LDA 모델 생성

```
library(MASS)
f.lda <- lda(Species~., data=data.train)
```

LDA 모델의 예측값 생성 및 확인

```
y.train.lda <- predict(f.lda, data.train)
y.test.lda <- predict(f.lda, data.test)
table(y.train.lda$class, data.train$Species)
```

```
##
##              versicolor virginica
##   versicolor         26         6
##   virginica           4        24
```

```
table(y.test.lda$class, data.test$Species)
```

```
##
##              versicolor virginica
##   versicolor         18        10
##   virginica           2        10
```

오차율 확인

```
(err.train.lda <- mean(y.train.lda$class != data.train$Species))
```

```
## [1] 0.1666667
```

```
(err.test.lda <- mean(y.test.lda$class != data.test$Species))
```

```
## [1] 0.3
```

SVM

패키지 로딩 및 SVM 모델 생성

```
library(e1071)
f.svm <- svm(Species~., data=data.train)
```

SVM 모델의 예측값 생성 및 확인

```
y.train.svm <- predict(f.svm, data.train)
y.test.svm <- predict(f.svm, data.test)
table(y.train.svm, data.train$Species)
```

```
##
## y.train.svm  versicolor virginica
##   versicolor         25         1
##   virginica           5        29
```

```
table(y.test.svm, data.test$Species)
```

```
##
## y.test.svm   versicolor virginica
##   versicolor         19         8
##   virginica           1        12
```

오차율 확인

```
(err.train.svm <- mean(y.train.svm != data.train$Species))
```

```
## [1] 0.1
```

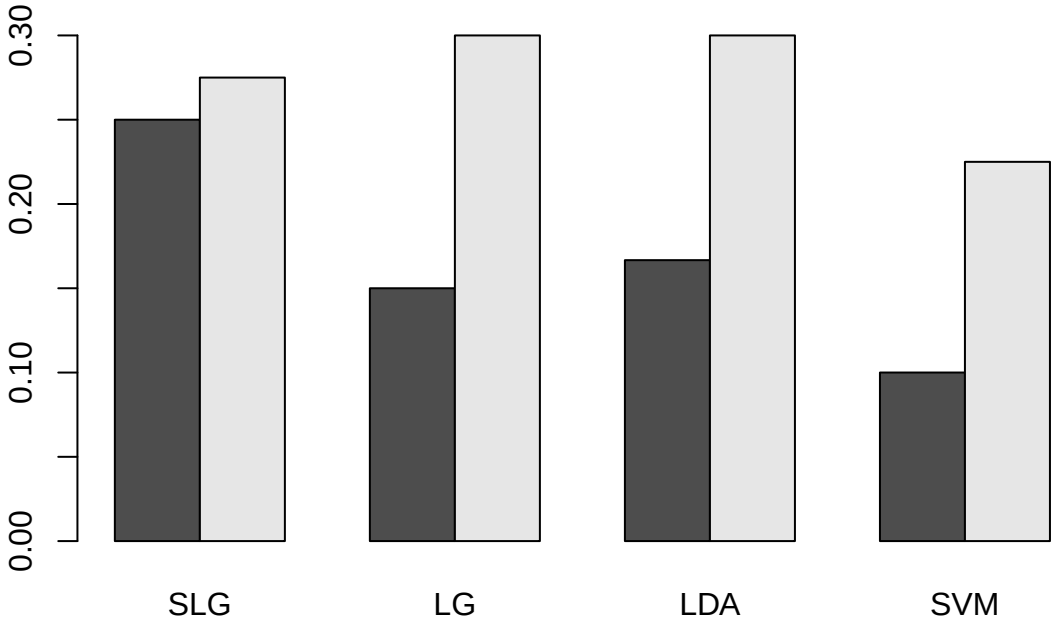
```
(err.test.svm <- mean(y.test.svm != data.test$Species))
```

```
## [1] 0.225
```

각 모델 비교

각 모델의 오차율 확인

```
ERR <- matrix(c(err.train.slg, err.test.slg,
                err.train.lg, err.test.lg,
                err.train.lda, err.test.lda,
                err.train.svm, err.test.svm), nrow=2)
colnames(ERR) <- c('SLG', 'LG', 'LDA', 'SVM')
rownames(ERR) <- c('train', 'test')
barplot(ERR, beside=TRUE)
```


0.00
0.10
0.20
0.30
SLG
LG
LDA
SVM