Homework#4 Problem#3-B

Arinjay Jain

October 22, 2020

```
classification.fun=function(train,test,i=4){

        temp    = train[,-1,with=F]
        temp$y[temp$y!=i]=0
        temp$y[temp$y==i]=1

        model = lm(y~.,data=temp)
        #summary(model)
        pred.train  = predict(model,temp[,-1,with=F])
        pred.test   = predict(model,test[,c(-1,-2),with=F])

        return(list(model))
    }


# Finally predicted final prediction
    pred.fun=function(total.model,train){

        pred.train =
                sapply(c(1:length(total.model)) ,
                function(x)
                predict(total.model[[x]],train[,c(-1,-2),with=F])
                ) %>% data.table

        colnames(pred.train) =
            paste("y=",c(1:length(total.model)),sep="")

        #apply( pred.train , 1 , sum )
        return(pred.train)
    }
```

```
library(MASS)
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.6.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
train_df <- read_csv("C:/Arinjay_Personal/Statistical Learning/Homework#4/vowel.train.txt",
    trim_ws = FALSE)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   row.names = col_double(),
##   y = col_double(),
##   x.1 = col_double(),
##   x.2 = col_double(),
##   x.3 = col_double(),
##   x.4 = col_double(),
##   x.5 = col_double(),
##   x.6 = col_double(),
##   x.7 = col_double(),
##   x.8 = col_double(),
##   x.9 = col_double(),
##   x.10 = col_double()
## )
```

```
#train_df <- data.frame(train_df)[-1]

test_df <- read_csv("C:/Arinjay_Personal/Statistical Learning/Homework#4/vowel.test.txt",
    trim_ws = FALSE)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
```

```
##   row.names = col_double(),
##   y = col_double(),
##   x.1 = col_double(),
##   x.2 = col_double(),
##   x.3 = col_double(),
##   x.4 = col_double(),
##   x.5 = col_double(),
##   x.6 = col_double(),
##   x.7 = col_double(),
##   x.8 = col_double(),
##   x.9 = col_double(),
##   x.10 = col_double()
## )
```

```
#test_df <- data.frame(test_df)[-1]


# Probability -> classification
class.pred.fun=function(tem){
    class.y = which( ( tem %in% max(tem) ) ==1 )
    return(class.y)
}
```

```
#misclassification , balance data, 11 classification
table(train_df$y)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11
## 48 48 48 48 48 48 48 48 48 48 48
```

```
# Do more models, 1 vs non-1
set.seed(100)
temp = sapply(c(1:11), function(x) classification.fun(train_df,test_df,i=x))
```

```
total.model = temp

pred.train = pred.fun(total.model,train_df)

# Probability classification change
pred.train.class = apply( pred.train,1,class.pred.fun)
# confusion matrix
t.train.matrix = table(train_df$y,pred.train.class)
t.train.matrix
```

```
##     pred.train.class
##       1  2  3  4  5  6  7  8  9 10 11
##   1  39  3  0  0  0  0  0  4  0  2  0
##   2  18 21  9  0  0  0  0  0  0  0  0
##   3   1  6 30  7  0  0  0  0  0  0  4
##   4   1  0  5 40  0  2  0  0  0  0  0
##   5   0  0  0  1 32  1 10  3  0  1  0
##   6   2  0  2 10 14  5 10  3  0  0  2
```

```
##   7   0  0  3  1 12  0 11 15  1  5  0
##   8   0  0  0  0  0  0  0 36  3  9  0
##   9   1  0  0  0  0  0  0 13 12 22  0
##   10  1  0  0  0  0  0  0  2  8 37  0
##   11 10  2  5  5  5  2  1  1  2  2 13
```

```
# Correct percent
train_acc <- sum( diag( t.train.matrix ) )/sum(t.train.matrix)
cat("Train accuracy: ", train_acc*100, "% \n")
```

```
## Train accuracy:  52.27273 %
```

```
# test before
pred.test  = pred.fun(total.model, test_df)
# Probability classification change
pred.test.class = apply( pred.test,1,class.pred.fun)
# confusion matrix
t.test.matrix = table(test_df$y,pred.test.class)

t.test.matrix
```

```
##     pred.test.class
##       1  2  3  4  5  6  7  8  9 10 11
##   1  41  0  1  0  0  0  0  0  0  0  0
##   2  25  5  9  0  0  0  0  0  0  3  0
##   3   4  5 21  8  0  4  0  0  0  0  0
##   4   0  0  4 26  6  6  0  0  0  0  0
##   5   0  0  0 15  9 12  3  3  0  0  0
##   6   1  0  6 13  9  8  2  0  0  3  0
##   7   0  5  0  6 18  3  0  1  2  7  0
##   8   0  0  0  0  4  0  0 18  1 19  0
##   9   0  0  2  0  0  0  0  6  3 31  0
##   10 12  0  4  0  0  0  0  0  4 22  0
##   11 11  1  8  9  0  1  0  0  2  9  1
```

```
# Correct percent
test_acc <- sum( diag( t.test.matrix ) )/sum(t.test.matrix)

cat("\n Misclassification error for the test data:", (1 -test_acc)*100, "%" )
```

```
##
##  Misclassification error for the test data: 66.66667 %
```

Using QDA

```
qda_fit <- qda(y~. -row.names, data = train_df)

pred <- predict(qda_fit, newdata = test_df)

classes <- pred$class
```

```
conf_mat <- table(classes, test_df$y)
conf_mat
```

```
##
## classes  1  2  3  4  5  6  7  8  9 10 11
##       1  37 18  9  0  0  0  0  0  0  2  0
##       2   4 22 13  2  0  0  0  0  0  4  1
##       3   0  1 12  3  0  0  0  0  0  0  0
##       4   0  0  5 12  0  1  0  0  0  0  2
##       5   0  0  0  5 16  0 11  0  0  0  0
##       6   0  0  2 17  7 22  1  0  0  0  1
##       7   0  0  0  2 19 14 22 15  3  4  2
##       8   0  0  0  0  0  0  0  6  1  0  0
##       9   1  1  1  0  0  0  3 21 38 21 15
##       10  0  0  0  0  0  0  0  0  0 11  1
##       11  0  0  0  1  0  5  5  0  0  0 20
```

```
# Correct percent
qda_acc <- sum(diag(conf_mat))/sum(conf_mat)

cat("\n Misclassification error for the test data using qda:", (1 -qda_acc)*100, "%" )
```

```
##
##  Misclassification error for the test data using qda: 52.81385 %
```

```
print("we are getting better result using QDA MASS R function with 52.81% Misclassification but is compu
```

```
## [1] "we are getting better result using QDA MASS R function with 52.81% Misclassification but is
computer program we are getting higer misclassification which is 66.667%"
```