January 2021

Assignment 1.2

```
# import data
data <- read.csv("default.csv")

#preparing data
data <- data[-c(1,3,4)]
data$AGE <- data$AGE/100

#splitting data
n=dim(data)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.4))
train=data[id,]
id1=setdiff(1:n, id)
set.seed(12345)
id2=sample(id1, floor(n*0.3))
valid=data[id2,]
id3=setdiff(id1,id2)
test=data[id3,]
```

```
#likelihood function
likelihood<-function(input_data,theta){

  Y <- as.matrix(input_data[,3])
  X <- input_data[,-3]
  X0 <- rep(1,nrow(input_data))
  X_new <- cbind(X0,X)
  n <- nrow(input_data)
  logl <- 0
  for (i in 1:length(Y)) {
    logl <- sum(logl + log(1 + exp(-Y[i]*t(theta)*X[i,])))
    }
  return(logl/n) # negative log-likelihood
}

parameter_a <- c(0,1,0)
likelihood_a <- likelihood(train,parameter_a)

parameter_b <- c(0,0,1)
likelihood_b <- likelihood(train,parameter_b)

parameter_c <- c(1,1,1)
likelihood_c <- likelihood(train,parameter_c)

df_loglik <- data.frame("Log-Likelihodd" =c(likelihood_a,
```

```
                                          likelihood_b,
                                          likelihood_c))

row.names(df_loglik) <- c("w=(0,1,0)","w=(0,0,1)","w=(1,1,1)")
knitr::kable(df_loglik)
```

	Log.Likelihodd
w=(0,1,0)	1.103252e+238
w=(0,0,1)	1.155479e+238
w=(1,1,1)	1.093353e+238

The log-likelihood value of a regression model is a way to measure the goodness of fit for a model. The higher the value of the log-likelihood, the better a model fits a data set. Logistic regression is a classical linear method for binary classification. By comparing, the log-likelihood values from the above table, it could be seen that the log-likelihood using the second parameter vector $(0, 0, 1)$ performed better; thus the target value would be better predicted by using the second parameter vector.

Assignment 1.3

```
optimal <- function(input_data){
  res <- optim( par = c(1,1,1), fn = likelihood, input_data = train)
  return(res)
}

best_par <- optimal(train)
best_par$par
```

```
## [1]   33.25926  271.92593 -197.49630
```

The decision boundary is:

$$33.25926 - 271.92593Sex - 97.49630Age$$

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-3
```

```
log_reg <- glm(default_payment ~ ., data = train, family=binomial)

pred_train <- predict(log_reg, newdata = train)
pred_test <- predict(log_reg, newdata = test)

misclass <- function(actual_val,fitted_val){
  confusion_matrix <- table(actual_val,fitted_val)
  n <- length(actual_val)
  error <- 1 - (sum(diag(confusion_matrix))/n)
  return(error)
}

train_error <- misclass(train$default_payment,pred_train)
test_error <- misclass(test$default_payment,pred_test)
```

```
df_error <- data.frame("Train error" = train_error,
                       "Test error" = test_error)

row.names(df_error) <- c("Misclassification Rates")
knitr::kable(df_error)
```

	Train.error	Test.error
Misclassification Rates	0.99375	0.9933333

From the above table it could be seen that both of the errors are pretty high. Both values are almost the same, the test error provided a slightly worse error compared to the test error.