TP2 MRR

Noah KWA MOUTOME - Victor TAN

2023-10-31

#IV. Cookies Study

Imports

```r
cookies_data <- read.csv("cookies.csv")
```

Features extraction

For each line (meaning, for each cookie), we will use the different spectral values to compute: the mean, the standard deviation, the slope, the minimum and the maximum.

```r
# Computation (mean, standard deviation, minimum and maximum)

cookies_data$mean <- rowMeans(cookies_data[, 2:701])
cookies_data$stDev <- apply(cookies_data[, 2:701], 1, sd)
cookies_data$min <- apply(cookies_data[, 2:701], 1, min)
cookies_data$max <- apply(cookies_data[, 2:701], 1, max)


# Computation (slope)

# Function: compute_slope
# @param: spectrum_values of a cookie (here, column 2 to 701)
# @return: slope of the spectrum curve for a cookie
compute_slope <- function(spectrum_values) {
  pos <- 1:length(spectrum_values)
  lm_model <- lm(spectrum_values ~ pos)
  slope <- coef(lm_model)[2]
  return(slope)
}

cookies_data$slope <- apply(cookies_data[, 2:701], 1, compute_slope)


# Display of the new columns
head(cookies_data[,702:706])
```

```
##        mean     stDev      min     max       slope
## 1 0.9851499 0.4111868 0.259270 1.73946 0.001914311
```

```
## 2 1.0355417 0.4123933 0.266864 1.66273 0.001898164
## 3 1.0010620 0.4025158 0.251654 1.60960 0.001860203
## 4 1.0280481 0.4040351 0.277777 1.63881 0.001861782
## 5 1.0655011 0.4158252 0.288328 1.70320 0.001910926
## 6 1.0840236 0.4262425 0.284625 1.74356 0.001967228
```

Regression model

Now, we have the different features of the spectra.

```r
# Only features and fat values are retrieved

cookies_features <- cookies_data[c("fat", "mean", "stDev", "slope", "min", "max")]
head(cookies_features)
```

```
##     fat      mean     stDev       slope      min     max
## 1 12.57 0.9851499 0.4111868 0.001914311 0.259270 1.73946
## 2 15.13 1.0355417 0.4123933 0.001898164 0.266864 1.66273
## 3 12.63 1.0010620 0.4025158 0.001860203 0.251654 1.60960
## 4 13.85 1.0280481 0.4040351 0.001861782 0.277777 1.63881
## 5 15.25 1.0655011 0.4158252 0.001910926 0.288328 1.70320
## 6 13.66 1.0840236 0.4262425 0.001967228 0.284625 1.74356
```

```r
X <- as.matrix(cookies_features[, -1]) # co-variables
y <- cookies_features$fat # target variable
```

```
## Le chargement a nécessité le package : Matrix
```

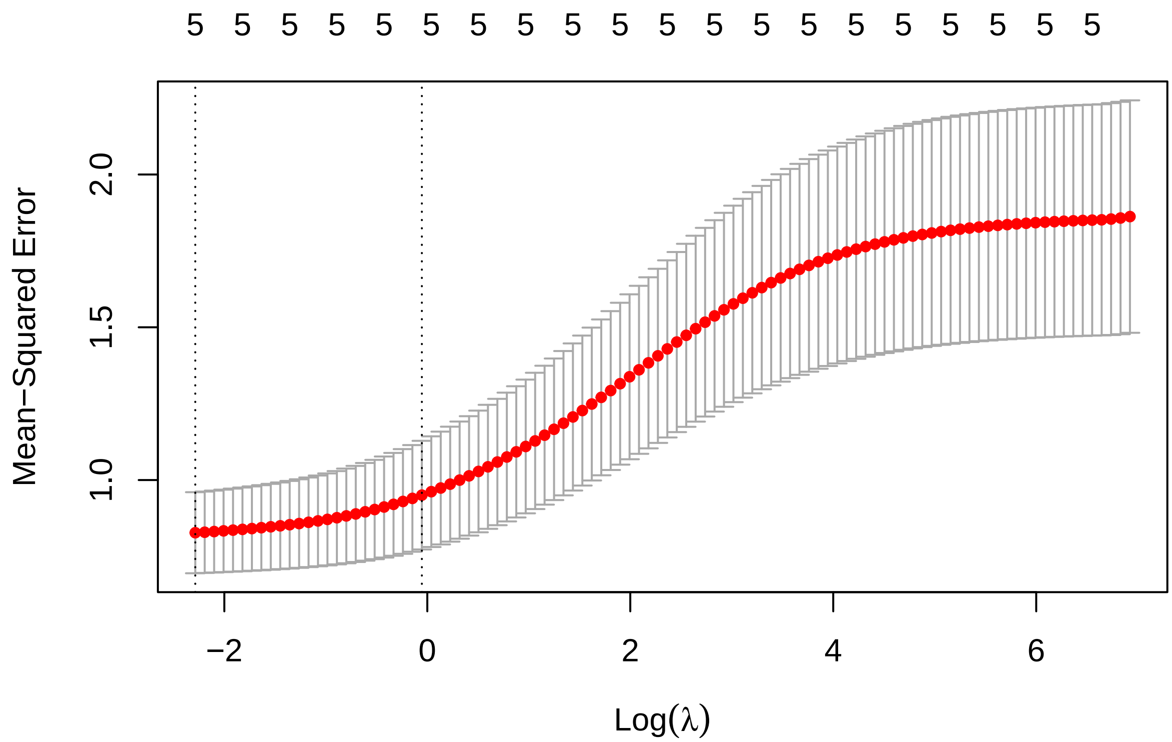
```
## Loaded glmnet 4.1-8
```

####Ridge regression

We're going to do the ridge regression first, using a cross validation k-fold to choose the best value for λ.

```r
# Cross validation
cv_ridge <- cv.glmnet(X, y, alpha=0)

plot(cv_ridge)
```




```
best_lambda <- cv_ridge$lambda.min # lambda that gives the lowest MSE
print(paste("The best value for lambda is", best_lambda))
```

```
## [1] "The best value for lambda is 0.101621626134997"
```

Now we have the best value for λ, we do another ridge regression with this parameter and there is its results :

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)    -3.1630322
## mean           12.2049944
## stDev           4.5091461
## slope       -1525.1388420
## min            24.2424028
## max            -0.6334611
```

```
predictions <- predict(best_model_ridge, newx = X)

# RMSE
rmse <- sqrt(mean((predictions - y)^2))
print(paste("RMSE ridge model :", rmse))
```

```
## [1] "RMSE ridge model : 0.820323805423126"
```

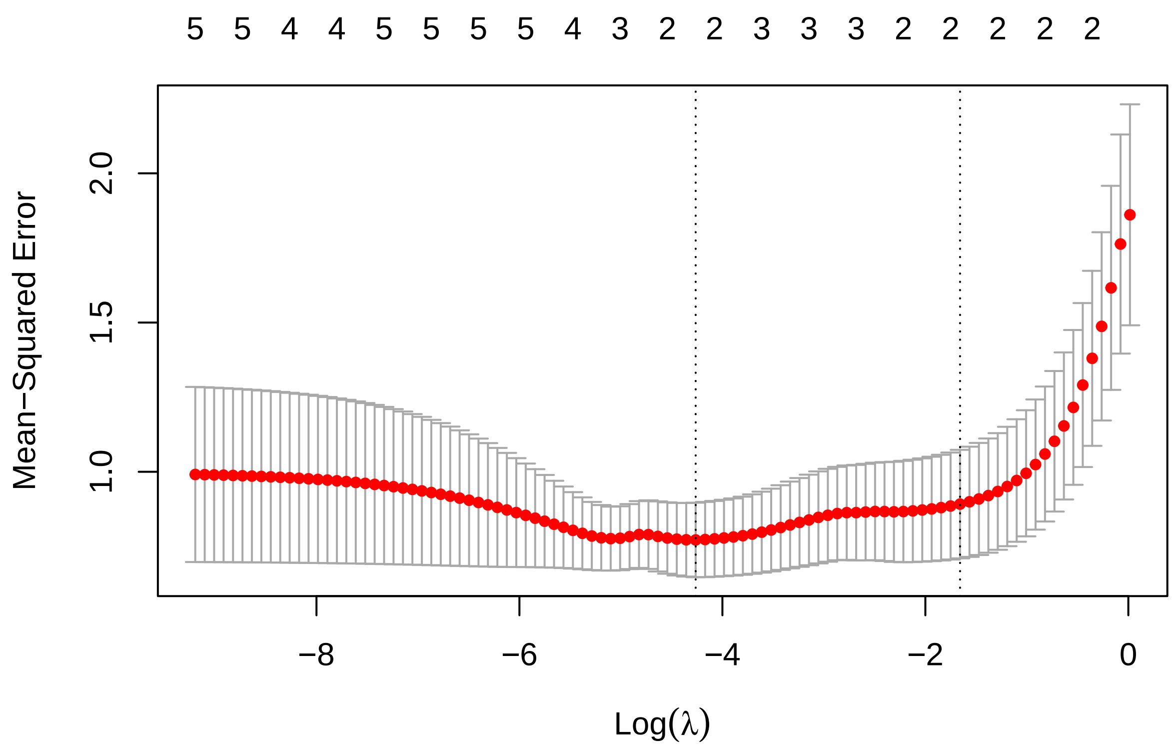
```r
# R^2
r_squared <- 1 - sum((y - predictions)^2) / sum((y - mean(y))^2)
print(paste("R^2 ridge model :", r_squared))
```

```
## [1] "R^2 ridge model : 0.617988425674367"
```

####Lasso regression

```r
cv_lasso <- cv.glmnet(X, y, alpha=1)

plot(cv_lasso)
```




```r
best_lambda <- cv_lasso$lambda.min # \(\lambda\) qui donne le plus bas MSE
print(paste("La meilleure valeur pour lambda est", best_lambda))
```

```
## [1] "La meilleure valeur pour lambda est 0.0140734405073241"
```

```r
best_model_lasso <- glmnet(X, y, alpha=1, lambda = best_lambda)
coef(best_model_lasso)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)    -2.389212
```

```
## mean           22.731721
## stDev               .
## slope       -3743.084744
## min                 .
## max                 .
```

```r
predictions <- predict(best_model_lasso, newx = X)

# RMSE
rmse <- sqrt(mean((predictions - y)^2))
print(paste("RMSE lasso model :", rmse))
```

```
## [1] "RMSE lasso model : 0.798500385801425"
```

```r
# R^2
r_squared <- 1 - sum((y - predictions)^2) / sum((y - mean(y))^2)
print(paste("R^2 lasso model :", r_squared))
```

```
## [1] "R^2 lasso model : 0.63804368957584"
```

####No penalization

```r
model_linear <- lm(y ~ X)

predictions_linear <- predict(model_linear, newdata = data.frame(X))

# RMSE
rmse_linear <- sqrt(mean((predictions_linear - y)^2))
print(paste("RMSE linear model :", rmse_linear))
```

```
## [1] "RMSE linear model : 0.706440945059225"
```

```r
# R^2
r_squared_linear <- 1 - sum((y - predictions_linear)^2) / sum((y - mean(y))^2)
print(paste("R^2 linear model :", r_squared_linear))
```

```
## [1] "R^2 linear model : 0.716692796197633"
```