HW4

Johnny Lydon

2024-03-22

HW4

```
## The following package(s) will be installed:
## - broom    [1.0.5]
## - car      [3.1-2]
## - caret    [6.0-94]
## - corrplot [0.92]
## - dplyr    [1.1.4]
## - nnet     [7.3-19]
## - purrr    [1.0.2]
## - readr    [2.1.5]
## - stringr  [1.5.1]
## - tidyr    [1.3.1]
## - torch    [0.12.0]
## These packages will be installed into "C:/Users/johnn/AppData/Local/R/win-library/4.3".
##
## # Installing packages --------------------------------------------------------
## - Installing dplyr ...                         OK [copied from cache in 0.48s]
## - Installing readr ...                         OK [copied from cache in 0.51s]
## - Installing purrr ...                         OK [copied from cache in 0.47s]
## - Installing stringr ...                       OK [copied from cache in 0.39s]
## - Installing tidyr ...                         OK [copied from cache in 0.6s]
## - Installing corrplot ...                      OK [copied from cache in 0.37s]
## - Installing nnet ...                          OK [copied from cache in 0.38s]
## - Installing broom ...                         OK [copied from cache in 0.49s]
## - Installing car ...                           OK [copied from cache in 0.37s]
## - Installing caret ...                         OK [copied from cache in 0.52s]
## - Installing torch ...                         OK [copied from cache in 0.61s]
## Successfully installed 11 packages in 6 seconds.

## Warning: package 'dplyr' was built under R version 4.3.3

## Warning: package 'readr' was built under R version 4.3.3

## Warning: package 'tidyr' was built under R version 4.3.3

## Warning: package 'purrr' was built under R version 4.3.3

## Warning: package 'stringr' was built under R version 4.3.3
```

```
## Warning: package 'corrplot' was built under R version 4.3.3

## Warning: package 'car' was built under R version 4.3.3

## Warning: package 'caret' was built under R version 4.3.3

## Warning: package 'torch' was built under R version 4.3.3

## Warning: package 'nnet' was built under R version 4.3.3

## Warning: package 'broom' was built under R version 4.3.3

##    dplyr    readr    tidyr    purrr  stringr corrplot      car    caret
##     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
##    torch     nnet    broom
##     TRUE     TRUE     TRUE
```

Question 1

```
#1.1

g <- function(x, y) {
  (x - 3)^2 + (y - 4)^2}

gradientg <- function(x, y) {
  gradientx = 2 * (x - 3)
  gradienty = 2 * (y - 4)
  return(c(gradientx, gradienty))
}

gradient <- gradientg(3, 4)
print(gradient)
```

```
## [1] 0 0
```

```
#Yes it mathces my expectations.
```

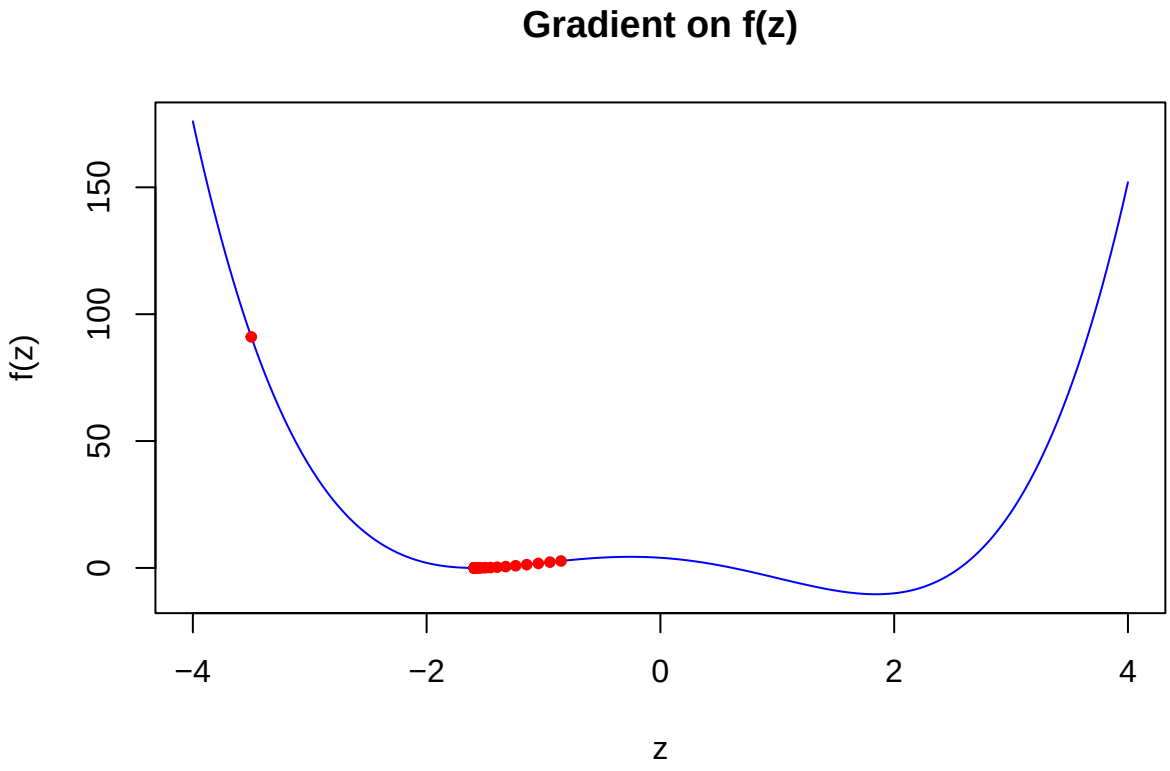
```
#1.2

install.packages("torch")
library(torch)

h <- function(u, v) {
  return((torch_dot(u, v))^3)}

u <- torch_tensor(c(-1, +1, -1, +1, -1, +1, -1, +1, -1, +1), dtype = torch_float())
v <- torch_tensor(c(-1, -1, -1, -1, -1, +1, +1, +1, +1, +1), dtype = torch_float())

gradient12 <- function(u, v) {
```

```
  h_value <- h(u, v)
  gradient <- grad(h_value, u)
  return(gradient)}

#Yes it matches my expectations.
```

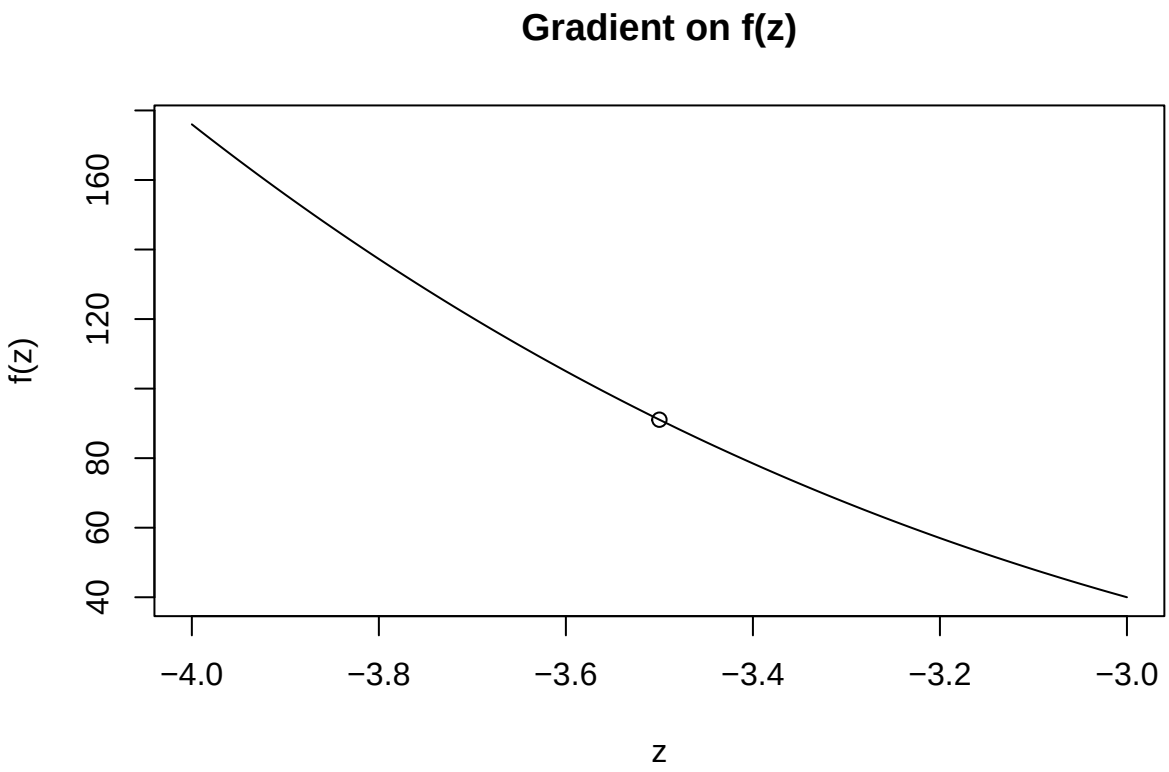
```
#1.3

f <- function(z) {
  z^4 - 6*z^2 - 3*z + 4}

dfdz <- function(z) {
  4*z^3 - 12*z - 3}

dfdz0 <- dfdz(-3.5)

print(dfdz0)
```

```
## [1] -132.5
```

```
#1.4

f <- function(z) {
  z^4 - 6*z^2 - 3*z + 4}

dfdz <- function(z) {
  4*z^3 - 12*z - 3}

z <- -3.5
eta <- 0.02
n <- 100

z_values <- c(z)

for (i in 1:n) {
  z <- z - eta * dfdz(z)
  z_values <- c(z_values, z)}

z_curve <- seq(from = -4, to = 4, length.out = 400)
f_curve <- f(z_curve)

plot(z_curve, f_curve, type = 'l', col = 'blue', xlab = 'z', ylab = 'f(z)', main = 'Gradient on f(z)')
points(z_values, f(z_values), col = 'red', pch = 20)
```




```r
#1.5

z2 <- -3.5
eta2 <- 0.03
n2 <- 100

z_values <- numeric(n2 + 1)
z_values[1] <- z2

for (i in 1:n2) {
  z2 <- z2 - eta2 * dfdz(z2)
  z_values[i + 1] <- z2}

curve(f, from = -4, to = -3, xlab = "z", ylab = "f(z)", main = "Gradient on f(z)")
points(z_values, f(z_values))
```



Question 2

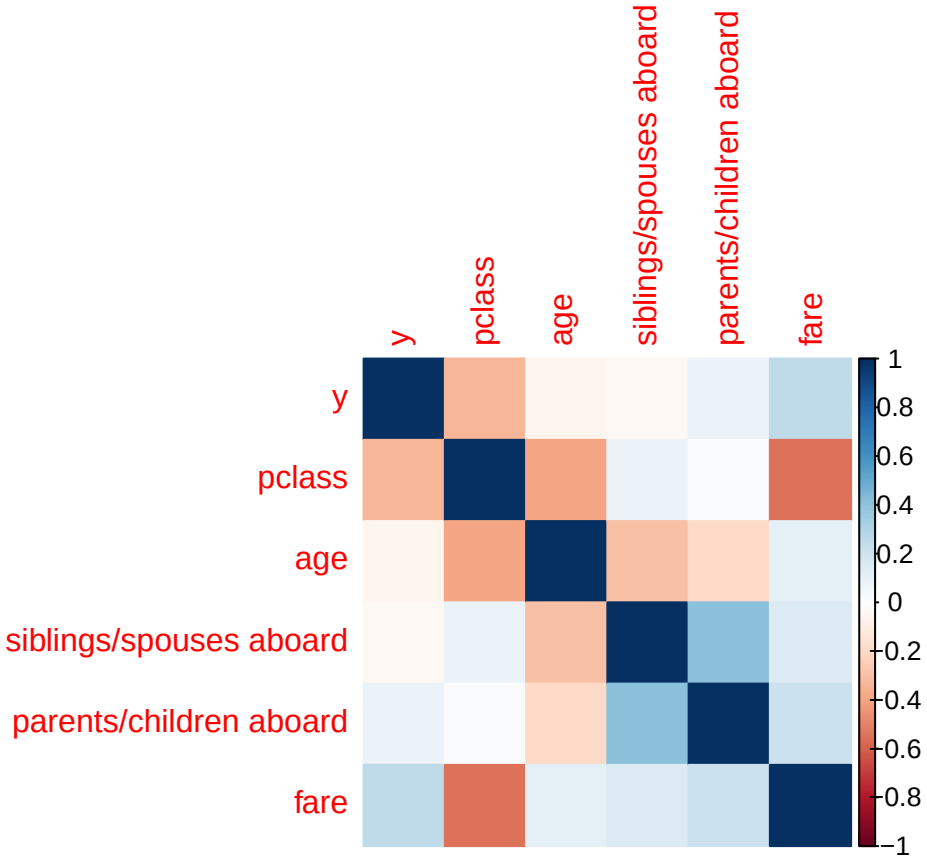
```
#2.1
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v tibble    3.2.1
## v lubridate 1.9.3
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x caret::lift()   masks purrr::lift()
## x car::recode()   masks dplyr::recode()
## x car::some()     masks purrr::some()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
url <- "https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv"

df <- read_csv(url) %>%
  mutate(across(where(is.character), as.factor)) %>%
  rename_with(tolower, everything()) %>%
  rename(y = survived)
```

```
## Rows: 887 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Name, Sex
## dbl (6): Survived, Pclass, Age, Siblings/Spouses Aboard, Parents/Children Ab...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#2.2

df %>% select_if(is.numeric) %>%
  cor() %>%
  corrplot(method = "color")
```




```
#2.3

full_model <- glm(y ~ pclass + sex + age + fare + `siblings/spouses aboard` + `parents/children aboard`

summary(full_model)
```

```
##
## Call:
## glm(formula = y ~ pclass + sex + age + fare + `siblings/spouses aboard` +
##     `parents/children aboard`, family = binomial, data = df)
```

```
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  5.297252   0.557409   9.503  < 2e-16 ***
## pclass                      -1.177659   0.146079  -8.062 7.52e-16 ***
## sexmale                     -2.757282   0.200416 -13.758  < 2e-16 ***
## age                         -0.043474   0.007723  -5.629 1.81e-08 ***
## fare                         0.002786   0.002389   1.166 0.243680
## `siblings/spouses aboard`  -0.401831   0.110712  -3.630 0.000284 ***
## `parents/children aboard`  -0.106505   0.118588  -0.898 0.369127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.77  on 886  degrees of freedom
## Residual deviance:  780.93  on 880  degrees of freedom
## AIC: 794.93
##
## Number of Fisher Scoring iterations: 5
```

```
#2.4
# The linear regression basically explains how likely it is that someone is going to get off the Titani
```

Question 3

```
#3.1

overview <- function(predicted, expected){
    accuracy <- sum(predicted == expected) / length(expected)
    error <- 1 - accuracy
    total_false_positives <- sum(predicted == 1 & expected == 0)
    total_true_positives <- sum(predicted == 1 & expected == 1)
    total_false_negatives <- sum(predicted == 0 & expected == 1)
    total_true_negatives <- sum(predicted == 0 & expected == 0)
    false_positive_rate <- total_false_positives / (total_false_positives + total_true_negatives)
    false_negative_rate <- total_false_negatives / (total_false_negatives + total_true_positives)
    return(
        data.frame(
            accuracy = accuracy,
            error=error,
            false_positive_rate = false_positive_rate,
            false_negative_rate = false_negative_rate
        )
    )
}
```

```
#3.2

summary(full_model)
```

```
##
```

```
## Call:
## glm(formula = y ~ pclass + sex + age + fare + `siblings/spouses aboard` +
##     `parents/children aboard`, family = binomial, data = df)
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 5.297252   0.557409   9.503  < 2e-16 ***
## pclass                     -1.177659   0.146079  -8.062 7.52e-16 ***
## sexmale                    -2.757282   0.200416 -13.758  < 2e-16 ***
## age                        -0.043474   0.007723  -5.629 1.81e-08 ***
## fare                        0.002786   0.002389   1.166 0.243680
## `siblings/spouses aboard` -0.401831   0.110712  -3.630 0.000284 ***
## `parents/children aboard` -0.106505   0.118588  -0.898 0.369127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.77  on 886  degrees of freedom
## Residual deviance:  780.93  on 880  degrees of freedom
## AIC: 794.93
##
## Number of Fisher Scoring iterations: 5
```

```r
#3.3

step_model <- step(full_model, direction = "backward")
```

```
## Start:  AIC=794.93
## y ~ pclass + sex + age + fare + `siblings/spouses aboard` + `parents/children aboard`
##
##                              Df Deviance     AIC
## - `parents/children aboard`   1   781.75  793.75
## - fare                        1   782.43  794.43
## <none>                            780.93  794.93
## - `siblings/spouses aboard`   1   796.85  808.85
## - age                         1   815.81  827.81
## - pclass                      1   847.84  859.84
## - sex                         1  1021.33 1033.33
##
## Step:  AIC=793.75
## y ~ pclass + sex + age + fare + `siblings/spouses aboard`
##
##                              Df Deviance     AIC
## - fare                        1   782.88  792.88
## <none>                            781.75  793.75
## - `siblings/spouses aboard`   1   801.59  811.59
## - age                         1   816.44  826.44
## - pclass                      1   852.19  862.19
## - sex                         1  1025.55 1035.55
##
## Step:  AIC=792.88
## y ~ pclass + sex + age + `siblings/spouses aboard`
##
```

```
##                              Df Deviance     AIC
## <none>                            782.88  792.88
## - `siblings/spouses aboard`  1   801.61  809.61
## - age                        1   818.41  826.41
## - pclass                     1   900.80  908.80
## - sex                        1  1031.86 1039.86
```

```
summary(step_model)
```

```
##
## Call:
## glm(formula = y ~ pclass + sex + age + `siblings/spouses aboard`,
##     family = binomial, data = df)
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  5.532066   0.504750  10.960  < 2e-16 ***
## pclass                      -1.265129   0.127021  -9.960  < 2e-16 ***
## sexmale                     -2.736487   0.195730 -13.981  < 2e-16 ***
## age                         -0.043697   0.007695  -5.679 1.36e-08 ***
## `siblings/spouses aboard` -0.407770   0.105197  -3.876 0.000106 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1182.77  on 886  degrees of freedom
## Residual deviance:  782.88  on 882  degrees of freedom
## AIC: 792.88
##
## Number of Fisher Scoring iterations: 5
```

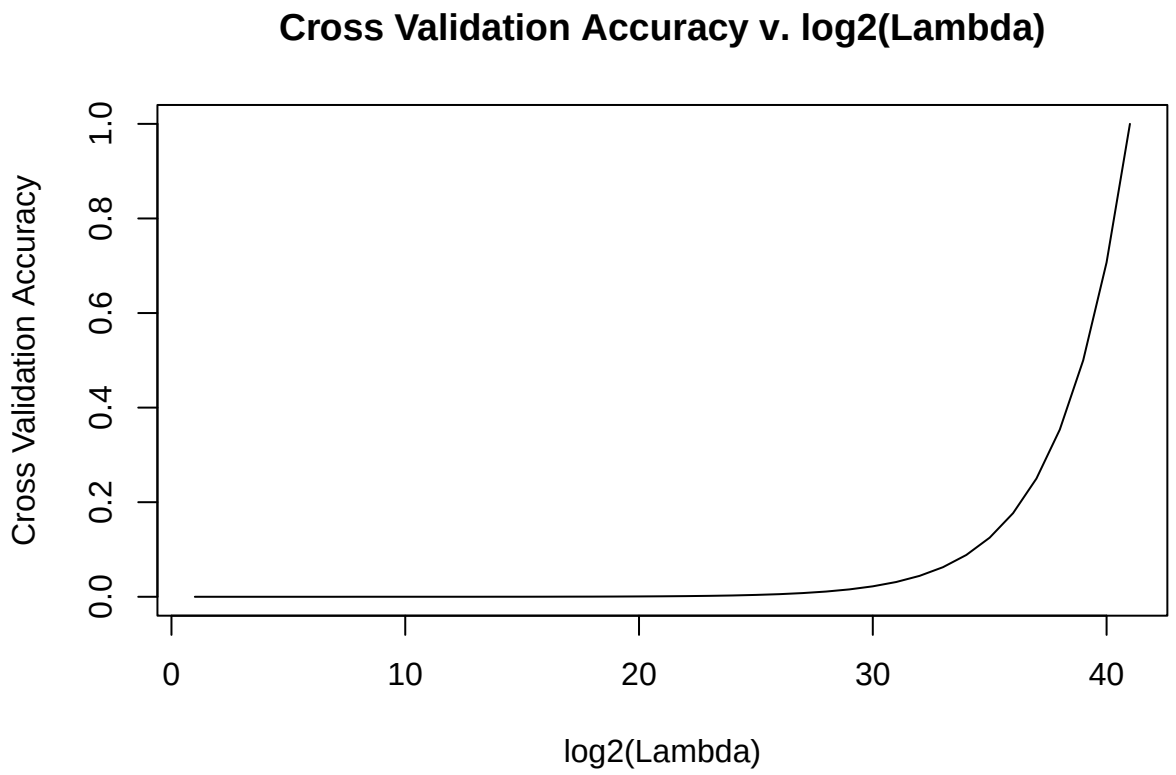
```
step_predictions <- predict(step_model, type = "response")
overview(step_predictions, df$y)
```

```
##   accuracy error false_positive_rate false_negative_rate
## 1        0     1                 NaN                 NaN
```

```
#3.4

controls <- trainControl(method="cv", number=5)

lasso_fit <- train(
  x = df[, -which(names(df) == "y")],
  y = df$y,
  method = "glmnet",
  trControl = controls,
  tuneGrid = expand.grid(
    alpha = 1,
    lambda = 2^seq(-20, 0, by = 0.5)
    ),
  family = "binomial"
)
```

```
## Warning in train.default(x = df[, -which(names(df) == "y")], y = df$y, method =
## "glmnet", : You are trying to do regression and your outcome only has two
## possible values Are you trying to do classification? If so, use a 2 level
## factor as your outcome column.

## Warning: Setting row names on a tibble is deprecated.

## Warning in storage.mode(xd) <- "double": NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning: Setting row names on a tibble is deprecated.

## Warning in storage.mode(xd) <- "double": NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning: Setting row names on a tibble is deprecated.

## Warning in storage.mode(xd) <- "double": NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning: Setting row names on a tibble is deprecated.

## Warning in storage.mode(xd) <- "double": NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning: Setting row names on a tibble is deprecated.

## Warning in storage.mode(xd) <- "double": NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.

## Warning: Setting row names on a tibble is deprecated.

## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
```

```r
plot(lasso_fit$results$lambda, lasso_fit$results$Accuracy, type = "l",
     xlab = "log2(Lambda)", ylab = "Cross Validation Accuracy",
     main = "Cross Validation Accuracy v. log2(Lambda)")
```




```r
#3.5

covariate_matrix <- model.matrix(full_model)[, -1]

X <- torch_tensor(covariate_matrix, dtype = torch_float())
y <- torch_tensor(df$y, dtype = torch_float())

logistic <- nn_module(
  initialize = function() {
    self$f <- nn_linear(in_features = 6, out_features = 1)
    self$g <- nn_dropout(p = 0.5)
  },
  forward = function(x) {
    x <- self$f(x)
    x <- self$g(x)
    torch_sigmoid(x)
    }
)

f <- logistic()
```

```r
Loss <- function(X, y, Fun){
  loss <- nn_binary_cross_entropy_with_logits()
  loss(Fun(X), y)
}

f <- logistic()
optimizer <- optim_adam(f$parameters(), lr = 0.01)

n <- 1000
for (i in 1:n) {
  optimizer$zero_grad()
  loss <- Loss(X, y, f$forward)
  loss$backward()
  optimizer$step()

  if (i %% 100 == 0) {
    cat("Iteration: ", i, " Loss: ", loss$item(), "\n")
  }
}

predicted_probabilities <- f(X) %>% as_array()
torch_predictions <- ifelse(predicted_probabilities > 0.5, 1, 0)

overview(torch_predictions, df$y)
```