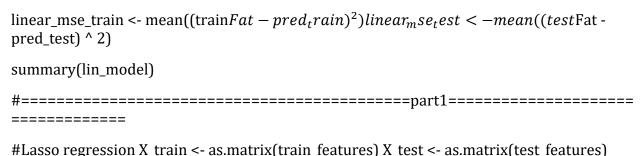
```
library(stats) library(caret) library(glmnet)  \begin{aligned} &\text{df} <\text{-read.csv("tecator.csv") n} <\text{-} \dim(df)[1] \text{ source("split_data.R") sets} <\text{-} \\ &\text{split_data(data\_frame = df, k = 2, seed = 12345, distributions = c(1,1)) train} <\text{-} \text{ sets}[[1]] \text{ test}} <\text{-} \\ &\text{-} \text{sets}[[2]] \end{aligned} \\ &\text{\#Channels as features train\_features} <\text{-} \text{ train}[1:(ncol(train)-3)] \text{ test\_features}} <\text{-} \\ &\text{test}[1:(ncol(test)-3)] \end{aligned} \\ &\text{\#fitting linear regression model lin\_model} <\text{-} \text{lm(train\$Fat} \sim\text{.} +0, data = train\_features}) \# +0 \text{ takes away the intercept} \\ &\text{\#predictions on the training data and test data pred\_train} <\text{-} \text{predict(lin\_model, train\_features})} \end{aligned}
```

Training and test errors



Fit LASSO regression model on the training data

lasso_model_train <- cv.glmnet(X_train, train\$Fat , alpha = 1) # Setting alpha = 1 for LASSO

LASSO Cost Function

 $lasso_cost_function <- function(X,y, beta, lambda) \{ N <- length(y) prediction <- X \%\% \ beta \ residuals <- y - prediction \ mse_term <- sum(residuals^2) / (2 N) l1_penalty <- lambda * sum(abs(beta)) cost <- mse_term + l1_penalty return(cost) \}$

Example usage of the cost function

optimal_lambda <- as.matrix(lasso_model_train\$lambda.min) optimal_beta <- coef(lasso_model_train, s = optimal_lambda) #The discrepancy in the dimensions suggests that the optimal_beta vector includes an additional element, #which typically corresponds to the intercept term. #In glmnet models, the intercept is included by default.

coefficients_optimal <- as.matrix(tail(optimal_beta, -1)) # Exclude the first element
(intercept)</pre>

Ensure that X_train_lasso and optimal_beta have compatible dimensions

if (ncol(X_train) == length(coefficients_optimal)) { cost <- lasso_cost_function(X_train,
train\$Fat, coefficients_optimal, optimal_lambda) cat("LASSO Cost Function Value:", cost, "")
} else { cat("Dimensions of X_train_lasso and optimal_beta are not compatible for matrix
multiplication.") }</pre>

Fit LASSO regression model on the training data

summary(lasso_model_train)

Plot the coefficient paths

plot(lasso_model_train\$glmnet.fit, "lambda", main = "LASSO Coefficient Paths", xlab =
"Log(lambda)", ylab = "Coefficients")

Identify the value of lambda for a model with only three features

desired_num_features <- 3 lambda_index <which.min(lasso_model_train_glmnet.fitdev.ratio > desired_num_features / ncol(X_train))

Add a vertical line at the selected lambda value

Fit Ridge regression model

ridge_model_train <- cv.glmnet(X_train, train\$Fat, alpha = 0) # Setting alpha = 0 for ridge

Plotting regression coefficients vs. $log(\lambda)$

Optimal λ and number of variables