Formative (2)

2024-11-17

```
#Loading libraries
library(ISLR)
library(glmnet)
```

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

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

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
#Removing rows that are missing values
Hitters <- na.omit(Hitters)

#Setting the variables
X <- as.matrix(Hitters[, -which(names(Hitters) == "Salary")])
y <- Hitters$Salary

train_index <- createDataPartition(y, p = 0.7, list = FALSE)

X_train <- X[train_index, ]
X_test <- X[-train_index, ]
y_train <- y[train_index]
y_test <- y[-train_index]

#Setting the seed
set.seed(2024)

#Ridge regression
ridge_model <- cv.glmnet(X_train, y_train, alpha = 0, nfolds = 10)
best_lambda_ridge <- ridge_model$lambda.min

# Prediction on test set using the best lambda
ridge_pred <- predict(ridge_model, s = best_lambda_ridge, newx = X_test)

# MSE for Ridge Regression
ridge_mse <- mean((ridge_pred - y_test)^2)
print(paste("Ridge MSE:", ridge_mse))
```

```
## [1] "Ridge MSE: 193107.236222895"
```

```
#Lasso regression
lasso_model <- cv.glmnet(X_train, y_train, alpha = 1, nfolds = 10)
best_lambda_lasso <- lasso_model$lambda.min
```

```r
#Prediction on tests using the best lambda
lasso_pred <- predict(lasso_model, s = best_lambda_lasso, newx = X_test)

# MSE for Lasso Regression
lasso_mse <- mean((lasso_pred - y_test)^2)
print(paste("Lasso MSE:", lasso_mse))
```

```
## [1] "Lasso MSE: 174480.035803666"
```

```r
#Lasso selected covariates
lasso_coef <- coef(lasso_model, s = best_lambda_lasso)

# Identifying selected features
selected_features <- which(lasso_coef[-1] != 0)
selected_feature_names <- rownames(lasso_coef)[-1][selected_features]

# Subset training and test sets using Lasso-selected features
X_train_lasso <- X_train[, selected_feature_names]
X_test_lasso <- X_test[, selected_feature_names]

# Combine selected predictors and response into data frames for linear model
train_lasso_data <- as.data.frame(cbind(X_train_lasso, y = y_train))
test_lasso_data <- as.data.frame(X_test_lasso)

# Linear model
lm_model <- lm(y ~ ., data = train_lasso_data)
```