Validation Using Various Cross Validation Methods

**The Issue**

We have babies weight dataset from which we have to develop a multivariate linear regression model and the apply different cross validation methods to find the test errors

Use the validation set method from pages 198-200 of the text to split the data randomly into two equal halves. Then, use one half for training the linear model and the other half as the test set to evaluate its performance.

To test the linear model using leave-one-out cross-validation (LOOCV), as described on pages 200-202 of the text, fit the model using all data points except one, and then evaluate its performance by predicting the omitted point. Repeat this process for each data point in the dataset and calculate the average error across all the predictions.

Use k-fold cross-validation with k = 10, as described on pages 203-206 of the text, to test the linear model. First, divide the dataset randomly into 10 equal-sized parts, or folds. Then, fit the model using 9 folds and use the remaining fold as the test set to evaluate its performance. Repeat this process 10 times, using a different fold as the test set each time. Finally, calculate the average error across all the 10 iterations.

**Findings**

The code used three cross-validation methods to test the multivariate linear regression model fitted to the Babiesweight.xls file. The results showed that the model has a moderate ability to predict birthweight based on the five predictor variables (Gestation, Age, Height, Weight, Smoke).

The R-squared values obtained from the validation set method, LOOCV, and 10-fold cross-validation mean were 0.0572, 0.0195, and 0.0421, respectively. Although these R-squared values are relatively low, they were consistent across all three cross-validation methods. This suggests that the model is not overfitting and is generalizing reasonably well to new data.

However, the low R-squared values also indicate that the model only explains a small proportion of the variation in birthweight. This means that there are likely other factors beyond the five predictor variables that influence birthweight as well. Therefore, while the model may be useful for predicting birthweight to some extent, it cannot be relied on entirely to predict birthweight accurately.

Dicussions

Appendix A: Method

Appendix B: Results

Appendix C: Code

We use the R language and write down the code in R studio.

Code:

library(readxl)

library(caret)

# Load data from Excel file

filex <- "C:/Users/DELL/Downloads/babies\_weight.xls"

babyx <- readxl::read\_xls(filex)

View(babyx)

X <- babyx[, c("Gestation", "Age", "Height", "Weight", "Smoke")]

y <- babyx$Birthweight

# Fit multivariate linear regression model

model1 <- lm(Birthweight~., data=babyx)

summary(model1)

# Use validation set method to split data into training and test sets

set.seed(42)

train\_idx <- createDataPartition(babyx$Birthweight, p = 0.5, list = FALSE)

train <- babyx[train\_idx,]

test <- babyx[-train\_idx,]

model2 <- lm(Birthweight~., data=train)

summary(model2)

y\_pred <- predict(model2, test)

print(paste("Validation set method R-squared:", summary(model2)$r.squared))

# Use LOOCV to test linear model

loocv <- trainControl(method = "LOOCV")

model3 <- train(Birthweight ~ ., data = babyx, method = "lm", trControl = loocv)

print(paste("LOOCV mean R-squared:", mean(model3$results$Rsquared)))

# Use k-fold cross-validation to test linear model

kfold <- trainControl(method = "cv", number = 10,

summaryFunction = defaultSummary)

model <- train(Birthweight ~ ., data = babyx, method = "lm", trControl = kfold)

print(paste("10-fold cross-validation mean R-squared:", mean(model$results$Rsquared)))­­­