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 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), 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, 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 means 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.

Discussion:

From the model we built, there few discussions which will be discussed.

The model is not very good at predicting outcomes, as shown by the low R-squared values obtained from two cross-validation methods. This suggests that the factors chosen may not be enough to fully explain the differences in birth weight.

The study found that smoking during pregnancy is linked to lower birth weight, emphasizing the need for smoking cessation during pregnancy. Longer gestational periods and taller mothers were associated with higher birth weights, supporting the significance of proper prenatal care and maternal health. However, the impact of age and weight on birth weight was not statistically significant after considering other variables in the model. Overall, the study suggests that the selected predictors may not be enough to fully explain the variation in birth weight.

The model's R-squared values obtained are extremely low. This implies that the model may not be suitable for the data and that other crucial factors may not be accounted for in the model. The model's inclusion of only five predictor variables may not be enough to fully comprehend the intricate relationships between the predictors and the outcome variable. It is possible that there are other significant predictors that were not considered in the model.

When interpreting the model's results on the correlation between predictor variables and birth weight, it is essential to be cautious due to several limitations and potential confounding factors that must be considered.

Appendix A: Method

This code reads data from an Excel file and separates predictor variables from the outcome variable. It uses a multivariate linear regression model with all five predictors to predict the outcome.

First, we employ the validation set method to split the data into training and test sets, with a 50/50 split. The model is fitted to the training set and used to make predictions on the test set. The R-squared value for the test set predictions is computed and printed to the console.

Next, the code uses leave-one-out cross-validation (LOOCV) to test the linear regression model. It creates a LeaveOneOut object and computes the R-squared value for each possible left-out observation. The average R-squared value over all left-out observations is then computed and printed to the console.

Finally, the code employs k-fold cross-validation (k=10) to test the model's predictive ability. It creates a KFold object and computes the R-squared value for each fold. The average R-squared value over all left-out observations is then computed and printed to the console.

Appendix B: Results

From the multivariate regression model, we built we got the result as follows,

```
Call:
lm(formula = Birthweight ~ ., data = babyx)

Residuals:
    Min      1Q  Median      3Q     Max
-65.231 -11.317   0.325  11.284  55.745

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 81.810363   7.947180  10.294  < 2e-16 ***
Gestation    0.012800   0.006830   1.874 0.061131 .
Age          0.070370   0.079456   0.886 0.375981
Height       0.525584   0.121922   4.311 1.76e-05 ***
Weight      -0.005831   0.004336  -1.345 0.178946
Smoke       -1.989031   0.561626  -3.542 0.000413 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.99 on 1230 degrees of freedom
Multiple R-squared:  0.03056,   Adjusted R-squared:  0.02661
F-statistic: 7.754 on 5 and 1230 DF,  p-value: 3.415e-07
```

And the final result from the three cross-validation methods simple validation method, LOOCV and k-fold method is as follows,

```
> print(paste("10-fold cross-validation mean R-squared:", mean(model$results$Rsquared)))
[1] "10-fold cross-validation mean R-squared: 0.0421335204770929"
> print(paste("Validation set method R-squared:", summary(model2)$r.squared))
[1] "Validation set method R-squared: 0.0572491003405421"
> print(paste("LOOCV mean R-squared:", mean(model3$results$Rsquared)))
[1] "LOOCV mean R-squared: 0.0195834679298251"
> print(paste("10-fold cross-validation mean R-squared:", mean(model$results$Rsquared)))
[1] "10-fold cross-validation mean R-squared: 0.0421335204770929"
```

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)))
```