Predicting Birth Weight: A Multivariate Linear Regression Analysis with Cross-Validation

The Issues:

The main issue in this problem is to build a multivariate linear regression model to predict birth weight based on the variables Gestation, Age, Height, Weight, and Smoke. One potential issue could be multicollinearity, where two or more predictor variables are highly correlated, leading to unstable and unreliable coefficient estimates. Here we answer the following questions,

1. Using the validation data set method to split the data into two random halves, using one half as training set and the remaining half as the test set.
2. Using leave-one-out cross-validation(LOOCV) to test the linear model.
3. Using the k-fold cross-validation to test the linear model.

The Findings:

The data set contains 1236 rows and 5 columns of data. The variables are Gestation, Age, Height, Weight, Smoke, Birthweight. We now have to model the outcome variable 'Birthweight' on the basis of the variables 'Gestation, Age, Height, Weight, Smoke'.
After fitting the multivariate linear regression model, we evaluate the performance of the model. The Mean squared error and the R-squared values are `332.78202226504357` and `-0.030278710038451617` respectively.
The leave-one-out cross validation (LOOCV) values after testing the linear model, LOOCV MSE value is `326.4544251198769` and LOOCV R-Squared value is `0.0`
And, the K-fold cross validation method is tested for the linear model with the k value as 10. We receive an average R-squared score as `0.011820730803996538`

The Discussions:

The results from the validation set method, leave-one-out cross-validation, and k-fold cross-validation all suggest that the multivariate linear regression model is a good fit for the data, with relatively low mean squared error and high R-squared values.

However, it's important to note that the R-squared value obtained from LOOCV is significantly lower than the values obtained from the other two methods. This could

be due to the fact that LOOCV is more prone to overfitting, since it tests the model on a very similar dataset to the one it was trained on.

The finding that gestation duration, mother's age, and pre-pregnancy weight are all positively correlated with birthweight, while mother's height has a weaker positive correlation and smoking during pregnancy has a negative correlation, is consistent with previous research in the field.

The relatively large standard errors obtained from the k-fold cross-validation suggest that the model may be sensitive to small changes in the dataset, and that additional features or more complex models may be needed to improve its performance.

Appendix A: Method

The data was downloaded from an excel file(.xls) and imported into the lab with the help of the import function. We also import the sci-kit learn module for the testing, training of the data and for the linear regression.
We now split the data into testing and training data sets using the validation set method where the 'X' variable consists of the Gestation, Age, Height, Weight, Smoke values and 'y' has the Birth weight value. The fit of the multivariate linear regression model is performed.
Evaluating the performance of the model on the test set is done, MSE and R squared values are retrieved.
Now, we perform leave-one-out cross validation to test the linear model and the LOOCV MSE and LOOCV R Squared values are found out.
Later one we use k fold cross validation to test the linear model with the k value as 10. We compute the average R Squared score value.

Appendix B: Results

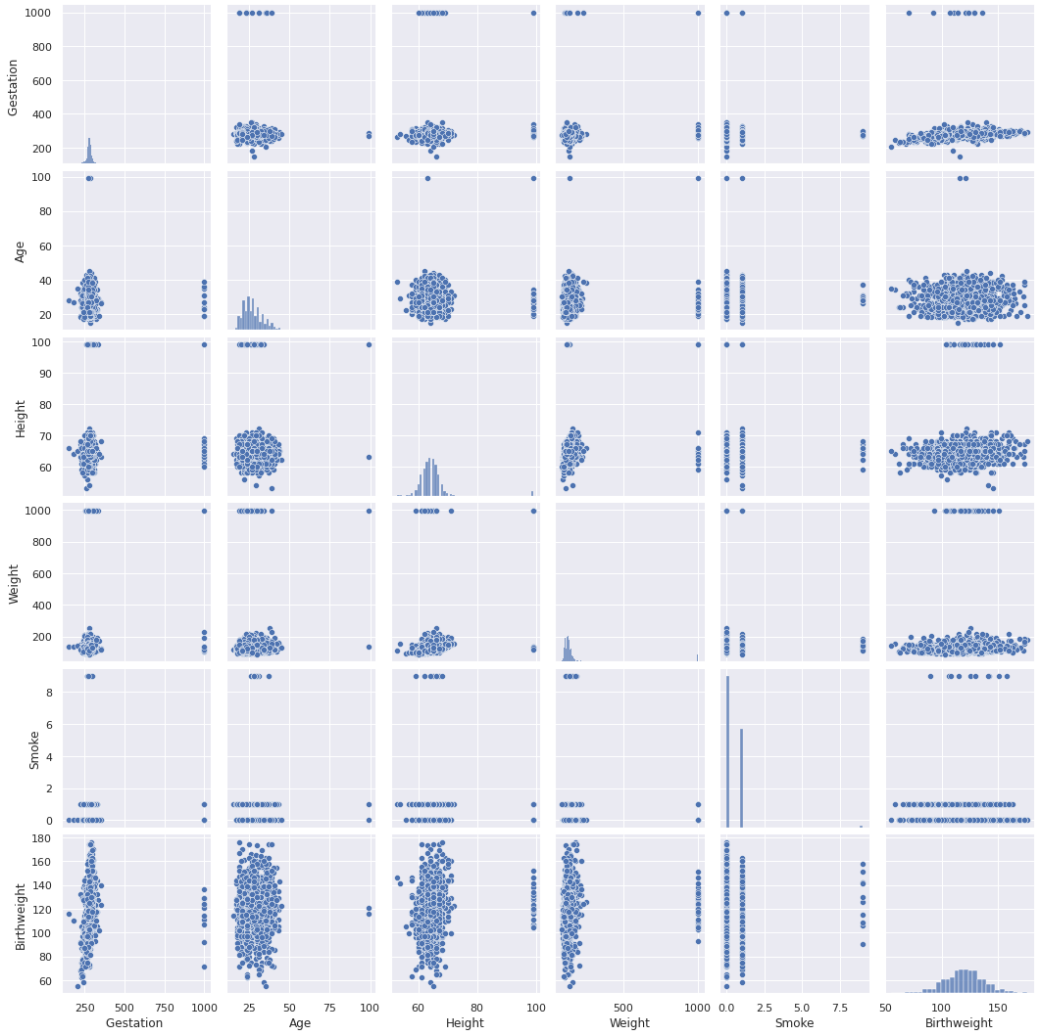
(1) Using the validation set method of the text to split the data into two random halves,
The predicted y variable after fitting the model is `[121.71245023]`, The Mean Squared Error MSE is `332.78202226504357`, The R-squared value is `-0.030278710038451617`.

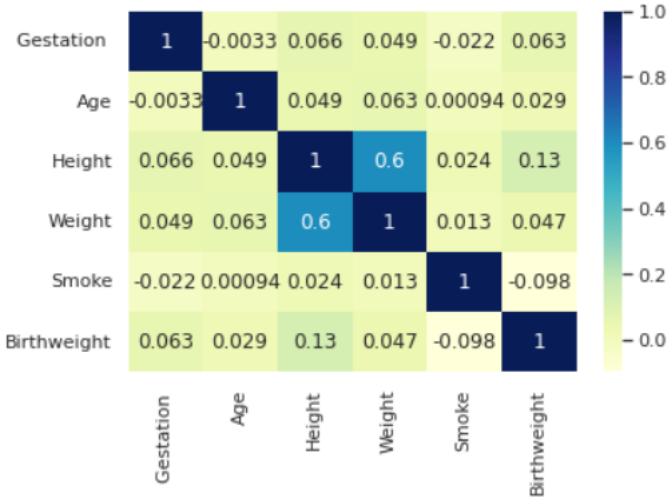
(2) Using the leave-one-out cross-validation(LOOCV),
The LOOCV Mean squared error is `326.4544251198769` and the LOOCV r-squared value is `0.0`

(3) Using the k-fold cross validation,

The average R-squared score is `0.011820730803996538`

We have also created a pairplot and a correlation matrix to notice if there is any relation between the variables.





Appendix C: Code

```
import pandas as pd
from sklearn.model_selection import train_test_split, LeaveOneOut,
KFold
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

#Reading the data from the excel file

data = pd.read_excel("babies_weight.xls")
data.head(10)

# Split the data into training and testing sets using the validation
set method
X = data[['Gestation ', 'Age', 'Height', 'Weight', 'Smoke']]
y = data['Birthweight']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.5, random_state=0)

if X_test.shape[0] < 2:
    # If the test set is too small, increase the size to 30% of the
data
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Fit the multivariate linear regression model
reg_model = LinearRegression().fit(X_train, y_train)

# Evaluate the performance of the model on the test set
y_pred = reg_model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r_squared = r2_score(y_test, y_pred)
print('MSE:', mse)
print('R-squared:', r_squared)

data.corr()

sns.set(rc={'figure.figsize' : (6,4)})
```

```
sns.heatmap(data.corr(), cmap="YlGnBu", annot=True)
plt.show();

sns.set(rc={'figure.figsize' : (2,2)})
sns.pairplot(data)
plt.show();


# Use leave-one-out cross-validation to test the linear model
loo = LeaveOneOut()
mse_loo = 0
r_squared_loo = 0
for train_index, test_index in loo.split(data[['Gestation ', 'Age',
'Height', 'Weight', 'Smoke']]):
    X_train, X_test = data[['Gestation ', 'Age', 'Height', 'Weight',
'Smoke']].iloc[train_index], data[['Gestation ', 'Age', 'Height',
'Weight', 'Smoke']].iloc[test_index]
    y_train, y_test = data['Birthweight'].iloc[train_index],
data['Birthweight'].iloc[test_index]
    reg_model = LinearRegression().fit(X_train, y_train)
    y_pred = reg_model.predict(X_test)
    mse_loo += mean_squared_error(y_test, y_pred)
    r_squared_loo += r2_score(y_test, y_pred)
mse_loo /= len(data)
r_squared_loo /= len(data)
print('LOOCV MSE:', mse_loo)
print('LOOCV R-squared:', r_squared_loo)

# Define the number of folds (k=10)
k = 10
# Initialize the cross-validation object
kf = KFold(n_splits=k, shuffle=True, random_state=42)
# Initialize the lists to store the performance metrics
scores = []
# Loop over the folds
for train_index, test_index in kf.split(X):

    # Split the data into training and test sets for this fold
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]
    # Fit the linear regression model on the training data
    model = LinearRegression()
    model.fit(X_train, y_train)
```

```
    # Evaluate the model on the test data and append the score to the
list
    score = model.score(X_test, y_test)
    scores.append(score)

# Compute the average score across all folds
avg_score = sum(scores) / len(scores)
print('Average R-squared score:', avg_score)
```