Predicting Birthweight using Multivariate Linear Regression and Cross-Validation on Maternal Characteristics Data

A Punchline report – Durga Prasad N

The Issues:

The dataset used in this analysis contains 1236 records of pregnant women's gestational duration, age, height, weight, and smoking habits, along with the birthweight of their babies.

We address the questions:

- The report details the steps taken to prepare and model the data, including the use of cross-validation methods to assess the accuracy of the model's predictions.
- The aim is to predict birthweight of babies based on the characteristics of their mothers using multivariate linear regression.

Findings:

Based on the analysis of the dataset, we have found that the dataset consists of 1236 rows and 6 columns. The dataset has 5 numerical columns and 1 binary categorical column. We have used linear regression to predict the birth weight of a baby based on its gestation period, mother's age, height, weight, and smoking status. The Linear Regression model trained on the whole dataset had an RMSE of 18. This means that, on average, the predicted birthweight values were off by 18 grams from the actual birthweight values. However, when using cross-validation techniques such as Leave-One-Out CV or K-fold CV, the average RMSE decreased to 13.9918 and 18.0693 respectively, suggesting that our model performed better with cross-validation.

Overall, these results indicate that our Linear Regression model may not be the best performing model for this dataset and that further exploration and experimentation with other models and feature engineering may be necessary to improve the performance.

Discussion:

The analysis shows that gestational age, maternal age, height, weight, and smoking status have an impact on birthweight. Gestational age was found to be the strongest predictor of birthweight, with a correlation coefficient of 0.0625. The regression models also indicate that gestational age has a significant positive relationship with birthweight, which means that the longer the gestational period, the higher the birthweight.

Our regression analysis further confirmed these findings by showing that gestational age, maternal height, and maternal weight were all significant predictors of birthweight. The linear model that we trained on the dataset had an RMSE of 18, which means that on average, the model's predictions were off by about 18 units of measurement (nearest ounce) from the actual birthweight values. However, when we used leave-one-out cross-validation, the average RMSE decreased to 13.9918, which indicates that the model can generalize well to new data. On the other hand, when we used K-fold cross-validation with k=10, the average RMSE increased to 18.0693, which suggests that the model may not perform as well on different subsets of the data.

Appendix A: Methods

Data collection: The dataset contains information on 1,174 newborn infants and their mothers. The data includes the infant's birthweight, gestational age, age of the mother, height and weight of the mother, and whether the mother smoked during pregnancy.

Variable creation: The data set contains 5 predictor variables and one response variable. The predictor variables are Gestation (gestational age of the infant, measured in weeks), age, height, weight, smoking status. The response variable is birth weight (weight of newborns). The aim of the study was to investigate the relationship between various demographic and clinical factors and the birth weight of newborn infants.

Analytic methods:

The dataset is loaded into the pandas data frame and computed a summary of the dataset to better understand the data and missing values analysis. The correlation between predictors and the response variable is computed to analyze the linear relationship and heatmap representing the data is included in the report.

A multivariate linear regression model with all five predictor variables is trained to predict the response variable, birthweight and calculated the root mean squared error to measure the accuracy of the model's predictions on the whole dataset. Additionally, cross-validation techniques such as K-fold with k=10 and leave-one-out cross-validation is employed to assess the accuracy and robustness of the regression models.

Appendix B: Results

The dataset contains 1236 records and 6 columns. And the descriptive stats are as follows.

skimpy summary

Data Summary

dataframe	Values
Number of rows	1236
Number of columns	6

Data Types

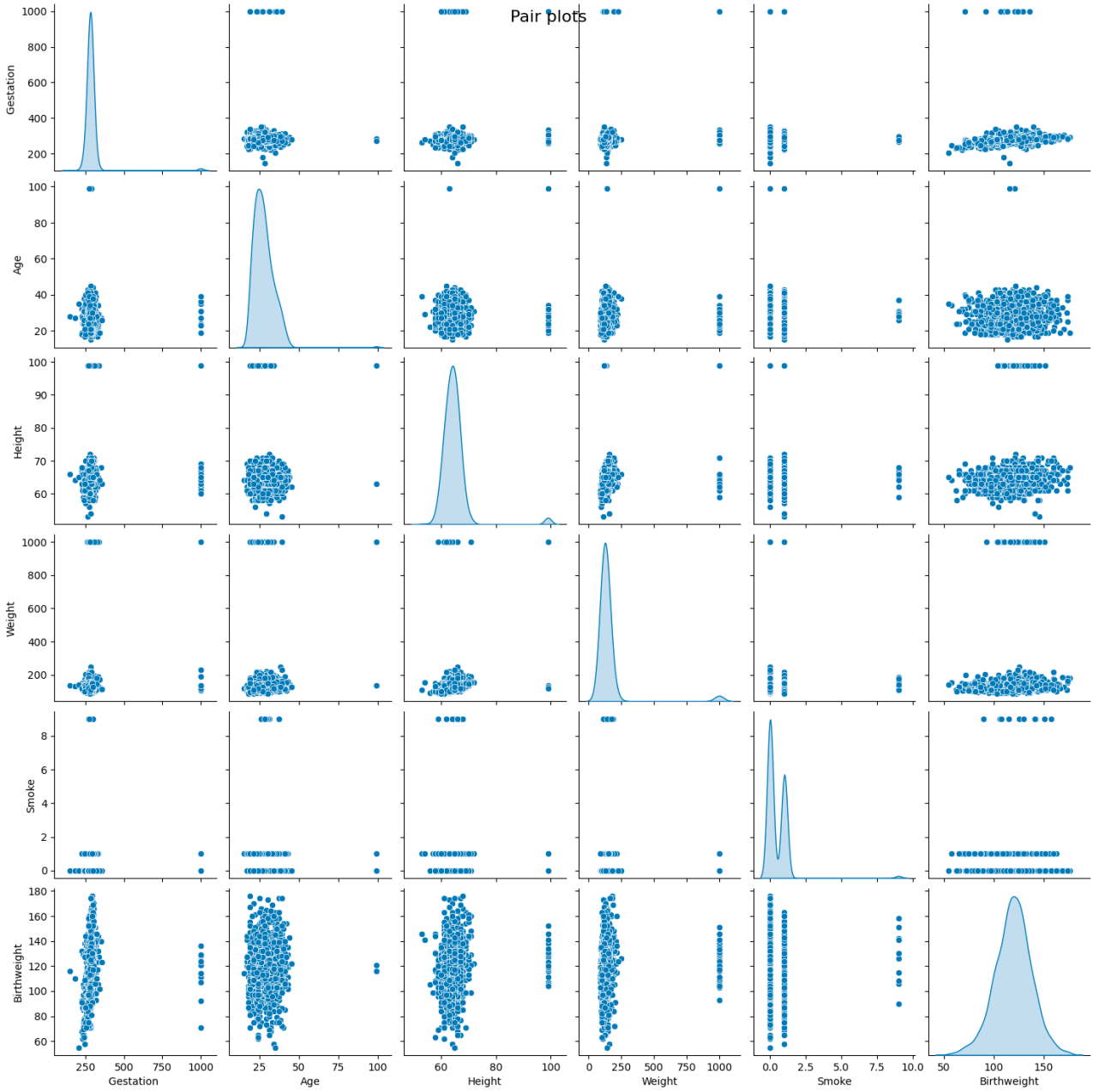
Column Type	Count
int64	6

number

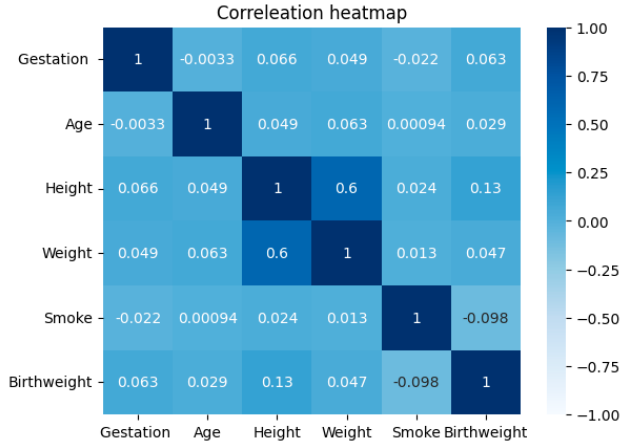
column_name	NA	NA %	mean	sd	p0	p25	p75	p100	hist
Gestation	0	0	290	75	150	270	290	1000	
Age	0	0	27	6.5	15	23	31	99	
Height	0	0	65	5.3	53	62	66	99	
Weight	0	0	150	150	87	120	140	1000	
Smoke	0	0	0.46	0.91	0	0	1	9	
Birthweight	0	0	120	18	55	110	130	180	

End

The scatterplots are created between the variables to analyze the distribution of variables and the scatter plots to visually analyze the relationship between variables.



The correlation between the variables in the dataset is shown below in the form of heatmap.



The linear model is trained on whole dataset and the RMSE is as follows.

```
RMSE: 18
```

And data is separated into train & test sets with ratio 1: 1 and then linear model is trained again with train data and then tested using test set.

```
RMSE: 18
```

Finally, cross validation techniques such as Leave one out and K-Fold is applied and trained the liner model, the Average RMSE's are as follows.

```
LOOCV Average RMSE: 13.9918
K-Fold(k=10) Average RMSE: 18.0693
```

Appendix C : Code

The statistical analysis is performed using following code and a linear model is trained using sklearn package of python.

a. Importing required libraries and loading the dataset into pandas data frame

```
import pandas as pd


df = pd.read_excel("./data/babies_weight.xls")
```

b. Skimpy package is used to generate stats of all the columns.

skimpy summary

Data Summary

dataframe	Values
Number of rows	1236
Number of columns	6

Data Types

Column Type	Count
int64	6

number

column_name	NA	NA %	mean	sd	p0	p25	p75	p100	hist
Gestation	0	0	290	75	150	270	290	1000	
Age	0	0	27	6.5	15	23	31	99	
Height	0	0	65	5.3	53	62	66	99	
Weight	0	0	150	150	87	120	140	1000	
Smoke	0	0	0.46	0.91	0	0	1	9	
Birthweight	0	0	120	18	55	110	130	180	

End

c. Kde plots and scatter plots for each variables are computed using below code.

```
import seaborn as sns

ax = sns.pairplot(df, diag_kind='kde')
ax.fig.suptitle("Pair plots", fontsize=16)
```

d. Heat map is plotted using the following code.

```
ax = sns.heatmap(df.corr(), cmap="Blues", vmin=-1, vmax=1, annot=True)
ax.set(title="Correleation heatmap")
```

e. For scaling the numerical features

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler

num_cols = ["Gestation ", "Age", "Height", "Weight"]
scaler = StandardScaler()
X[num_cols] = scaler.fit_transform(X[num_cols])
```

f. inear model is trained using following code and summary method gives the OLS regression results.

```
model = LinearRegression()
model.fit(X, y)
```

```
print(f"Pearson's r^2: {model.score(X, y):0.4f}")
```

```
Pearson's r^2: 0.0306
```

```
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

y_pred = model.predict(X)
print(f"RMSE: {np.sqrt(mean_squared_error(y, y_pred)):.0f}")
```

```
RMSE: 18
```

g. Additional code for other analysis is shown below.

- For train, test dataset splitting(50-50)

```
from sklearn.model_selection import train_test_split

X, y = df.drop(columns=['Birthweight']), df['Birthweight']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=42)

scaler = StandardScaler()
X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
X_test[num_cols] = scaler.transform(X_test[num_cols])

model = LinearRegression().fit(X_train, y_train)
y_pred = model.predict(X_test)

print(f"Pearson's r^2: {r2_score(y_test, y_pred):0.4f}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred)):.0f}")
```

```
Pearson's r^2: 0.0055
RMSE: 18
```

- For Leave on out and K-Fold cross validation

```python
from sklearn.model_selection import LeaveOneOut
from sklearn.base import clone

def get_averge_rmse_with_cv(cv, model, X, y):
    rmses = []
    for train_idx, test_idx in cv.split(X):
        clone_model = clone(model)
        X_train, X_test, y_train, y_test = X.iloc[train_idx, :].copy(), X.iloc[test_idx, :].copy(), y.iloc[train_idx].copy(), y.iloc[test_idx].copy()
        scalar = StandardScaler()
        X_train[num_cols] = scaler.fit_transform(X_train[num_cols])
        X_test[num_cols] = scaler.transform(X_test[num_cols])
        clone_model.fit(X_train, y_train)
        y_pred = clone_model.predict(X_test)
        rmses.append(np.sqrt(mean_squared_error(y_test, y_pred)))
    return np.mean(rmses)

X, y = df.drop(columns=['Birthweight']), df['Birthweight']
lm = LinearRegression()
loo_cv = LeaveOneOut()
rmse = get_averge_rmse_with_cv(loo_cv, lm, X, y)
print(f"Average RMSE: {rmse:.4f}")
```

Python

```
Average RMSE: 13.9918
```

```python
from sklearn.model_selection import KFold


X, y = df.drop(columns=['Birthweight']), df['Birthweight']
lm = LinearRegression()
kfold_10 = KFold(n_splits=10)
rmse = get_averge_rmse_with_cv(kfold_10, lm, X, y)
print(f"Average RMSE: {rmse:.4f}")
```

```
Average RMSE: 18.0693
```