Calories Burnt Prediction Clean Documentation

This document explains the Calories Burnt Prediction notebook. The project uses XGBoost with XGBRegressor as the single model for training and prediction. The code included below is extracted exactly from the notebook and wrapped to avoid page overflow.

1 Overview

This project predicts calories burnt during exercise using XGBoost with XGBRegressor.

The notebook includes dataset loading, merging, exploratory analysis, visualization, preprocessing,

training the XGBoost model, and evaluation based on regression metrics.

2 Libraries used

pandas

numpy

matplotlib

seaborn

sklearn model selection and preprocessing utilities if used

xgboost XGBRegressor

3 Workflow steps

1 Load and merge datasets

2 Perform exploratory data analysis

3 Visualize correlations and distributions

4 Prepare feature matrix X and target vector Y

5 Split into training and testing sets

6 Train XGBRegressor on training data

7 Evaluate model performance on test data

8 Predict calories burnt for new input samples

4 Important variables

calories

exercise data

df merged dataset

X Y

xgb_model or model

prediction

5 Notes

Only XGBoost is used as the predictive model in this notebook

All code below is included exactly as in the notebook with line wrapping applied

6 Complete notebook code

Code Cell 1

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn import metrics
```

Code Cell 2

```
calories = pd.read_csv('/content/drive/MyDrive/Data Science/Projects/16 Calories Burnt
Prediction/calories.csv')
excercise_dataset = pd.read_csv('/content/drive/MyDrive/Data Science/Projects/16 Calories
Burnt Prediction/exercise.csv')
calories.head()
```

Code Cell 3

```
excercise_dataset.head()
```

Code Cell 4

```
calories_data = pd.concat([excercise_dataset, calories["Calories"]], axis=1)
calories_data.head()
```

Code Cell 5

```
#checking no. of rols and cols
calories_data.shape
```

Code Cell 6

```
#more info for the dataframe
calories_data.info()
```

Code Cell 7

```
#checking the null values
calories_data.isnull().sum()
```

Code Cell 8

```
#get some statistical measures of the data
calories_data.describe()
```

Code Cell 9

```
sns.set()
```

Code Cell 10

```
calories_data['Gender'].value_counts()
```

Code Cell 11

```
#plotting the gender
sns.countplot(x=calories_data['Gender'])
```

Code Cell 12

```
#finding the distribution of age column
sns.distplot(calories_data['Age'])
plt.show()
```

Code Cell 13

```
sns.distplot(calories_data['Height'])
```

Code Cell 14

```
sns.distplot(calories_data['Weight'])
```

Code Cell 15

```
#converting the text data to numerical values
calories_data.replace({
    'Gender':{'male':0, 'female':1}
}, inplace=True)
```

Code Cell 16

```
calories_data.head()
```

Code Cell 17

```
correlation = calories_data.corr()
```

Code Cell 18

```
plt.figure(figsize=(10,10))
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot=True,
annot_kws={'size':8}, cmap='Blues')
```

Code Cell 19

```
x = calories_data.drop(columns = ['Calories', 'User_ID'], axis=1)
y = calories_data['Calories']
x.head()
```

Code Cell 20

```
y.head()
```

Code Cell 21

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=2)
x.shape, x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Code Cell 22

```
#loading the model
model = XGBRegressor()
```

Code Cell 23

```
#training the model
model.fit(x_train, y_train)
```

Code Cell 24

```
training_data_prediction = model.predict(x_train)
training_data_prediction
```

Code Cell 25

```
test_data_prediction = model.predict(x_test)
test_data_prediction
```

Code Cell 26

```
mae = metrics.mean_absolute_error(y_test, test_data_prediction)
mae
```

Code Cell 27

```
input_data = x_test.iloc[1]
predictions = model.predict([input_data])
```

```
predictions
```

Code Cell 28

```
x_test
```

Code Cell 29

```
y_test
```

Code Cell 30

```
# (Empty code cell)
```

Author Credit

Author Satyam Gajjar