

Name: Atharva Chaudhari

Roll No: 42

Experiment No. 1

Aim:

Implementation of Linear and Logistic Regression with Performance Analysis and Visualization.

Dataset Source:

The dataset used in this experiment is the Calories Burned Dataset.

 Kaggle Source Link:

<https://www.kaggle.com/code/jeeelsheikh/calories-burnt-prediction>.

Dataset Description:

The dataset contains information related to physical activity and calories burned.

1. Features (Independent Variables):

- **Gender** – Male/Female (converted to numeric)
- **Age** – Age of the person
- **Height** – Height in cm
- **Weight** – Weight in kg
- **Duration** – Duration of exercise (minutes)
- **Heart_Rate** – Heart rate during exercise
- **Body_Temp** – Body temperature during exercise

2. Target Variable:

- Calories – Number of calories burned

3. Dataset Characteristics:

- Structured tabular dataset (CSV format)
- Contains both categorical and numerical data
- Suitable for:
 - a) Regression (predicting calories)
 - b) Classification (high/low calorie burn)

Mathematical Formulation of the Algorithms:

1. Linear Regression:

Linear Regression predicts a continuous value using the equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

Where:

- y = Predicted value (Calories)
- x_i = Independent variables
- β_i = Coefficients
- β_0 = Intercept

Loss Function:

$$MSE = \frac{1}{n} \sum (y_{actual} - y_{predicted})^2$$

Used in my code:

```
mean_squared_error(y_test, y_pred)
```

2. Logistic Regression:

Used for binary classification.

Logistic Function (Sigmoid):

$$P(y = 1) = \frac{1}{1 + e^{-z}}$$

Where:

$$z = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$

It outputs probability between 0 and 1.

In My code:

```
y_binary = np.where(y > y.mean(), 1, 0)
```

Algorithm Limitations:

a) Linear Regression:

1. Assumes linear relationship
2. Sensitive to outliers
3. Cannot handle non-linear patterns well
4. Multicollinearity affects performance

b) Logistic Regression:

1. Works only for binary classification
2. Assumes linear decision boundary
3. Sensitive to imbalanced datasets

Methodology / Workflow:

Step 1: Data Upload:

- CSV file uploaded using Google Colab

Step 2: Data Preprocessing:

- Cleaned column names
- Converted Gender into numeric format
- Applied one-hot encoding for categorical features

Step 3: Data Splitting:

- Train-Test Split (80%-20%)

Step 4: Correlation Analysis:

- Heatmap used to analyze feature relationships

Step 5: Linear Regression:

- Model trained using `LinearRegression()`
- Predictions generated
- Performance evaluated using MSE
- Visualization:
 - a) Scatter Plot (Actual vs Predicted)
 - b) Line Plot
 - c) Box Plot

Step 6: Logistic Regression:

- Converted target into binary
- Model trained using `LogisticRegression()`
- Evaluated using Accuracy Score

Performance Analysis:

1. Linear Regression:

Evaluation Metric:

- **Mean Squared Error (MSE)**

Interpretation:

- Lower MSE → Better model performance
- Shows prediction error magnitude

Visualization:

- Scatter Plot → Shows closeness of predicted vs actual
- Line Plot → Comparison trend
- Box Plot → Distribution of calories

2. Logistic Regression:

Evaluation Metric:

- **Accuracy Score**

Interpretation:

- Higher Accuracy → Better classification
- Indicates percentage of correct predictions

Hyperparameter Tuning:

1. Linear Regression:

Basic Linear Regression has fewer hyperparameters.

Possible tuning:

- Regularization (Ridge/Lasso)
- Feature scaling

2. Logistic Regression:

Hyperparameters:

- `max_iter`
- Regularization parameter `C`
- Solver selection

Code:

```
# Upload CSV file
from google.colab import files
uploaded = files.upload()

# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.metrics import mean_squared_error, accuracy_score

# Load dataset
file_name = list(uploaded.keys())[0]
df = pd.read_csv(file_name)

# Clean column names
df.columns = df.columns.str.strip()

# Convert categorical data to numeric
if 'Gender' in df.columns:
    df['Gender'] = df['Gender'].map({'male': 1, 'female': 0})

df = pd.get_dummies(df, drop_first=True)

print("Dataset Loaded Successfully")
print(df.head())
```

```

# 1 CORRELATION HEATMAP

plt.figure(figsize=(8,6))
sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()

# LINEAR REGRESSION

X = df.iloc[:, :-1]
y = df.iloc[:, -1]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

linear_model = LinearRegression()
linear_model.fit(X_train, y_train)

y_pred = linear_model.predict(X_test)

print("\nLINEAR REGRESSION")
print("Mean Squared Error:", mean_squared_error(y_test, y_pred))

# 2 SCATTER PLOT (Actual vs Predicted)

plt.figure()
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("Scatter Plot: Actual vs Predicted")
plt.show()

# 3 LINE PLOT (Actual & Predicted)

plt.figure()
plt.plot(range(len(y_test)), y_test.values, label="Actual Values")
plt.plot(range(len(y_pred)), y_pred, label="Predicted Values")
plt.xlabel("Test Data Index")
plt.ylabel("Calories")
plt.title("Line Plot: Actual vs Predicted")
plt.legend()
plt.show()

# 4 BOX PLOT (Calories Distribution)

plt.figure()
sns.boxplot(y=y)
plt.title("Box Plot of Calories")
plt.ylabel("Calories")
plt.show()

# LOGISTIC REGRESSION

y_binary = np.where(y > y.mean(), 1, 0)

X_train, X_test, y_train, y_test = train_test_split(
    X, y_binary, test_size=0.2, random_state=42
)

logistic_model = LogisticRegression(max_iter=1000)
logistic_model.fit(X_train, y_train)

y_pred = logistic_model.predict(X_test)

print("\nLOGISTIC REGRESSION")
print("Accuracy:", accuracy_score(y_test, y_pred))

print("\n✅ Experiment Completed Successfully")

```

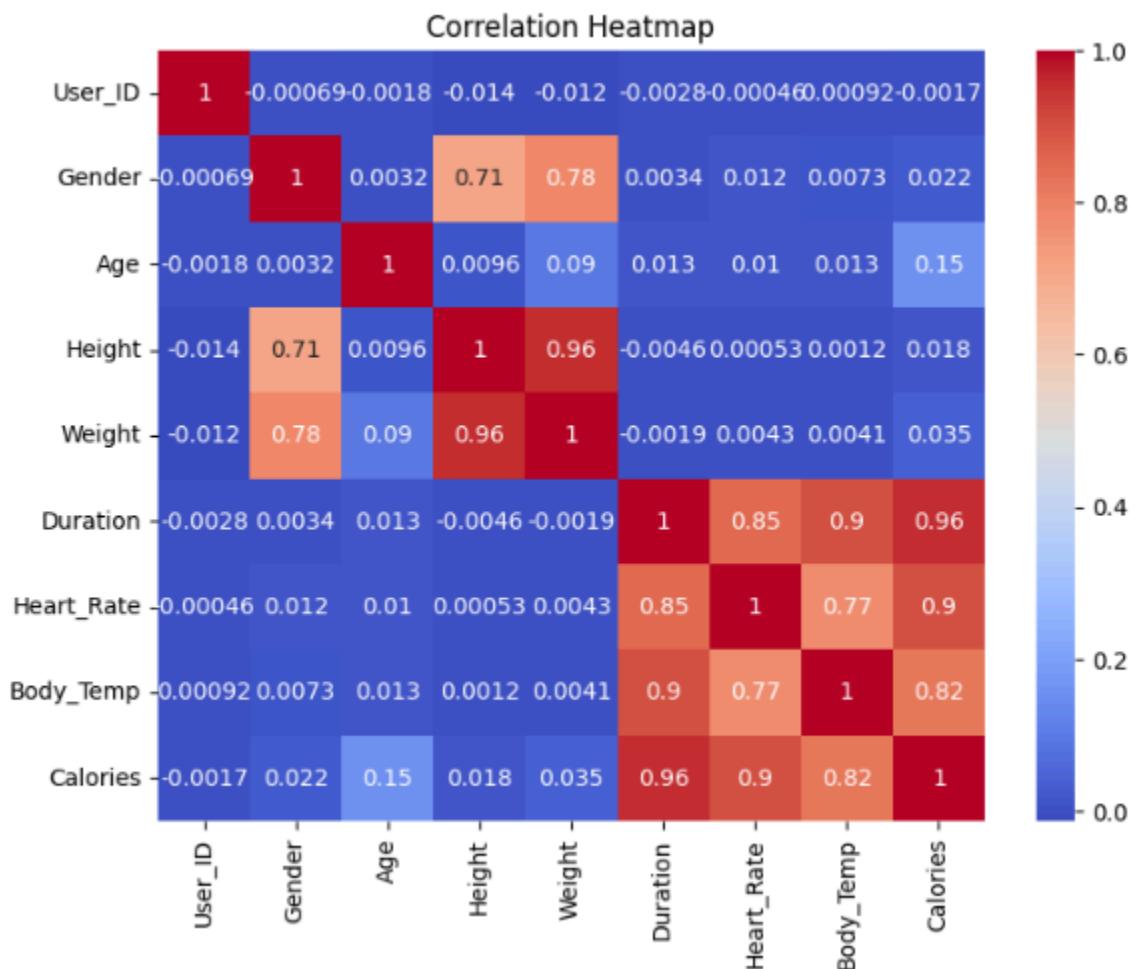
OUTPUT:

1.

```
[Choose Files] No file chosen          Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving calories.csv to calories (1).csv
Dataset Loaded Successfully
   User_ID  Gender  Age  Height  Weight  Duration  Heart_Rate  Body_Temp \
0  14733363      1   68    190.0    94.0       29.0     105.0      40.8
1  14861698      0   20    166.0    60.0       14.0      94.0      40.3
2  11179863      1   69    179.0    79.0       5.0      88.0      38.7
3  16180408      0   34    179.0    71.0      13.0     100.0      40.5
4  17771927      0   27    154.0    58.0       10.0      81.0      39.8

   Calories
0    231.0
1     66.0
2     26.0
3    71.0
4    35.0
```

2.

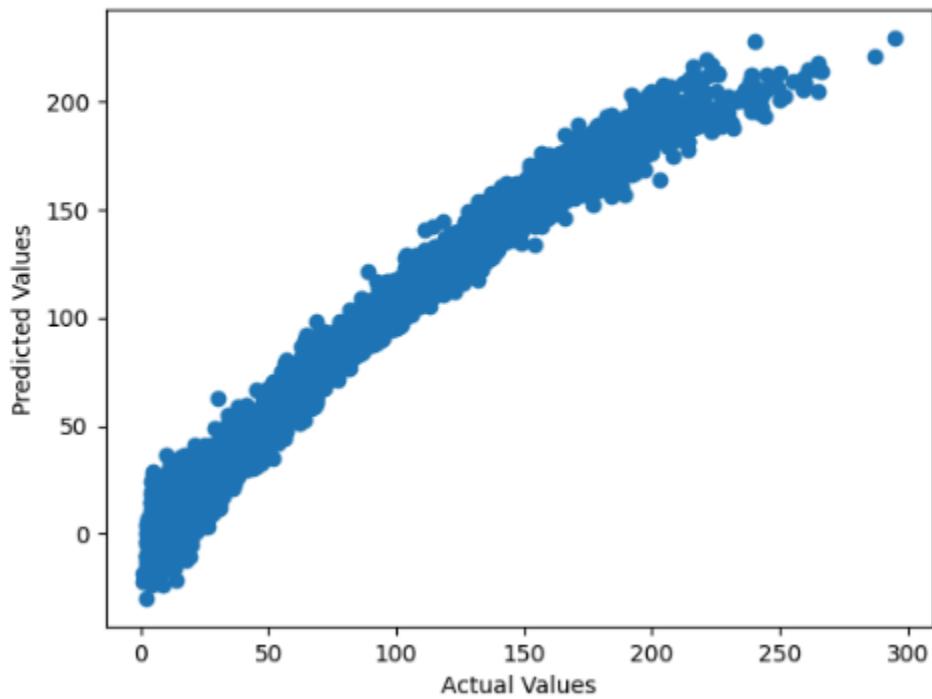


3.

LINEAR REGRESSION

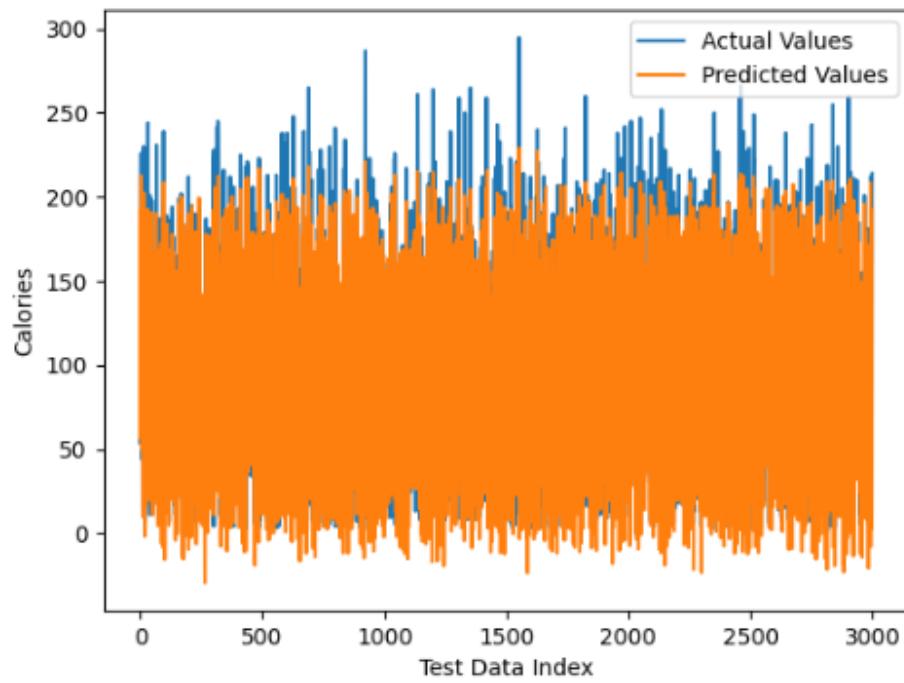
Mean Squared Error: 132.0675823406652

Scatter Plot: Actual vs Predicted

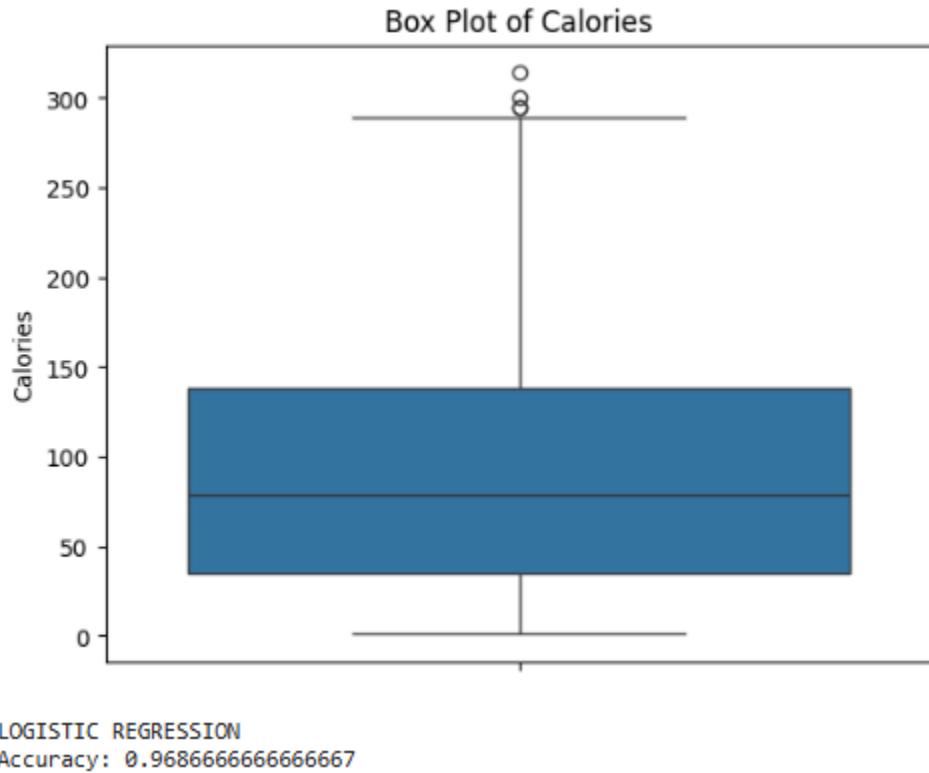


4.

Line Plot: Actual vs Predicted



5.



Conclusion:

In this experiment, both Linear Regression and Logistic Regression were implemented successfully.

Linear Regression effectively predicted continuous calorie values and its performance was evaluated using Mean Squared Error.

Logistic Regression classified calorie burn into high and low categories with good accuracy.

Visualization techniques such as heatmap, scatter plot, line plot, and box plot helped in understanding feature relationships and model behavior.