SOMAIYA
VIDYAVIHAR
K J Somaiya College of Science & Commerce

Somaiya
TRUST

INDEX

INTRODUCTION

In today's world, where people are leading busy lives with changes in their lifestyle and work commitments, it has become difficult to prioritize regular physical activity to maintain good health. The lack of physical activity and unhealthy food habits can lead to various health issues, including obesity. To maintain a healthy lifestyle, it is crucial to balance diet and exercise, and knowing calorie intake and burn is essential.

When people hear the word “calories”, they typically associate it with food and weight loss. However, calories are units of heat energy, measured as the amount of energy required to raise 1 gram of water by 1°C. While this measurement can be used to assess energy-releasing systems unrelated to the human body, in the context of the human body, it refers to the amount of energy required to perform a task. Food contains varying amounts of energy, with each item having a distinct calorie count. During exercise, the body's temperature and heart rate increase as carbohydrates are broken down into glucose and converted to energy with the help of oxygen.

To predict the amount of energy burned during exercise, various parameters such as duration, average heart rate, temperature, height and weight can be considered.

Calculating daily calorie burn. Being able to work out how many calories are burned each day is essential to any person looking to maintain, lose, or gain weight. Knowing what factors contribute to calorie burning can help a person alter their diet or exercise program to accommodate the goal.

Burning calories regularly will keep your pre-existing health conditions in check and reduce the risk of their severity. Exercising can improve the blood flow to the brain, reducing the risk of strokes. Burning calories can make your blood vessels healthier, lowering the risk of cardiovascular diseases

Methodology

The statistical methods used in this project are

1. Descriptive Statistics :
   Descriptive statistics refers to a set of methods used to summarize and describe the main features of a dataset, such as its central tendency, variability, and distribution. These methods provide an overview of the data and help identify patterns and relationships.

2. Testing of Hypothesis :
   A statistical method used to assess whether there is enough evidence in a sample of data to support a particular hypothesis about a population. It involves formulating two competing hypothesis: a) Null hypothesis (H0) and the alternative hypothesis.

3. Multiple Regression Analysis:
   Multiple regression is an extension of simple linear regression. It is used when we want to predict the value of a variable (i.e. dependent variable) based on the value of two or more other variables (independent variables).

4. Ordinary Least Square Regression:
   Ordinary Least Squares regression (OLS) is a common technique for estimating coefficients of linear regression equations which describe the relationship between one or more independent quantitative variables and a dependent variable.

5. Residual Analysis:
   Residual analysis, also known as residual diagnostics, is a technique used in statistical modelling and regression analysis to assess the quality and validity of a model's assumptions and to detect any patterns or systematic deviations in the model's residuals.

6. Variable Selection Method:
   Variable selection refers to the process of choosing the most relevant variables to include in a regression model. They help to improve model performance and avoid over fitting.

7. Autocorrelation:
   Autocorrelation is a measure of how strongly the output of a model is influenced by its lagged variables. For a model to be a good fit Autocorrelation should not be present. Therefore for detecting autocorrelation we can use Durbin-Watson (D-W) test for our model.

8. Heteroscedasticity:
   Heteroscedasticity means unequal scatter of the residuals over the range of fitted values. For a regression model to be a good fit for the data Heteroscedasticity should not be present i.e. it should be Homoscedastic.

9. Multicollinearity:
   In regression analysis, Multicollinearity occurs when two or more predictor variables are highly correlated to each other, it can cause problems when we fit the model and interpret our results. Therefore for detecting Multicollinearity we use the method of high pair wise correlation among regressors.

AIM

To predict calories burned depending on various factors

Objectives

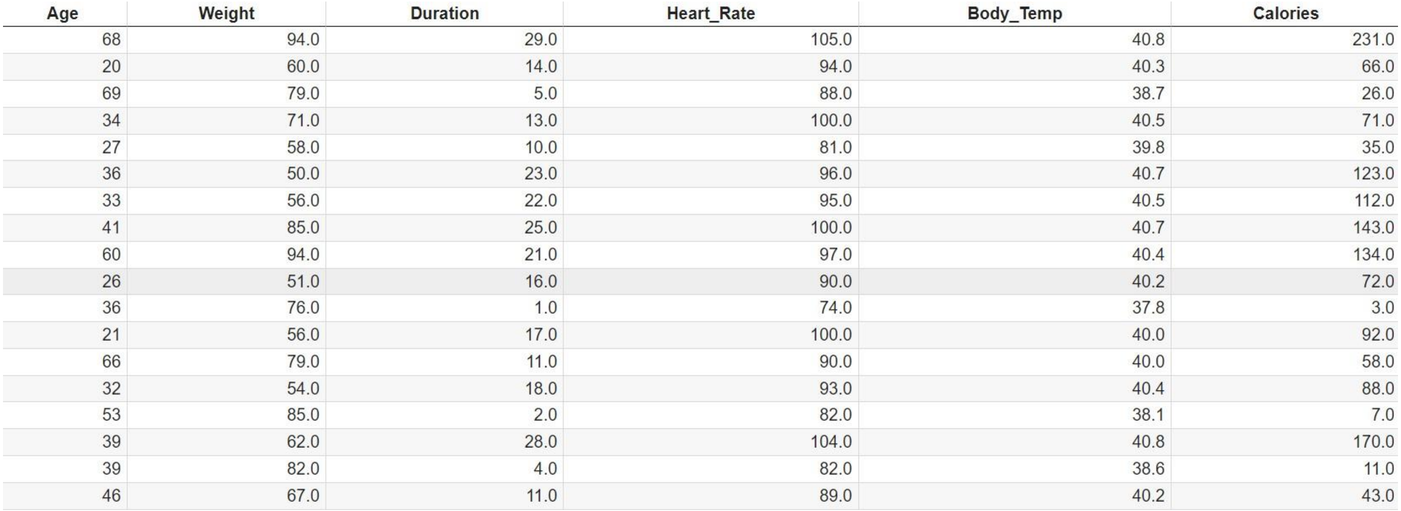
- To fit a linear regression model for the output calories burned depending upon several input variables.
- To check the overall significance and individual significance.
- To find out most relevant independent variables using variable selection method.
- To find out the relationship between the study variable and the predictors.
- To check whether Autocorrelation is present or not.
- To check whether Heteroscedasticity is present or not.
- To test whether Multicollinearity is present or absent in the data.
- To compare the linear model results with XGBoost results.

DATA

Data collection is an essential part of any statistical analysis and machine learning projects, as the quality of the data has a significant impact on the performance of the resulting model. **In this project, data is collected from Kaggle**, a popular platform for data scientists and machine learning practitioners to access and share datasets.

Link for the dataset: https://www.kaggle.com/datasets/fmendes/fmendesdat263xdemos

Once the dataset was collected, it was uploaded to **Google Colab**, a cloud-based platform for data analysis and machine learning. In this work, the dataset contained over **15,000 records** and **7 variables**.



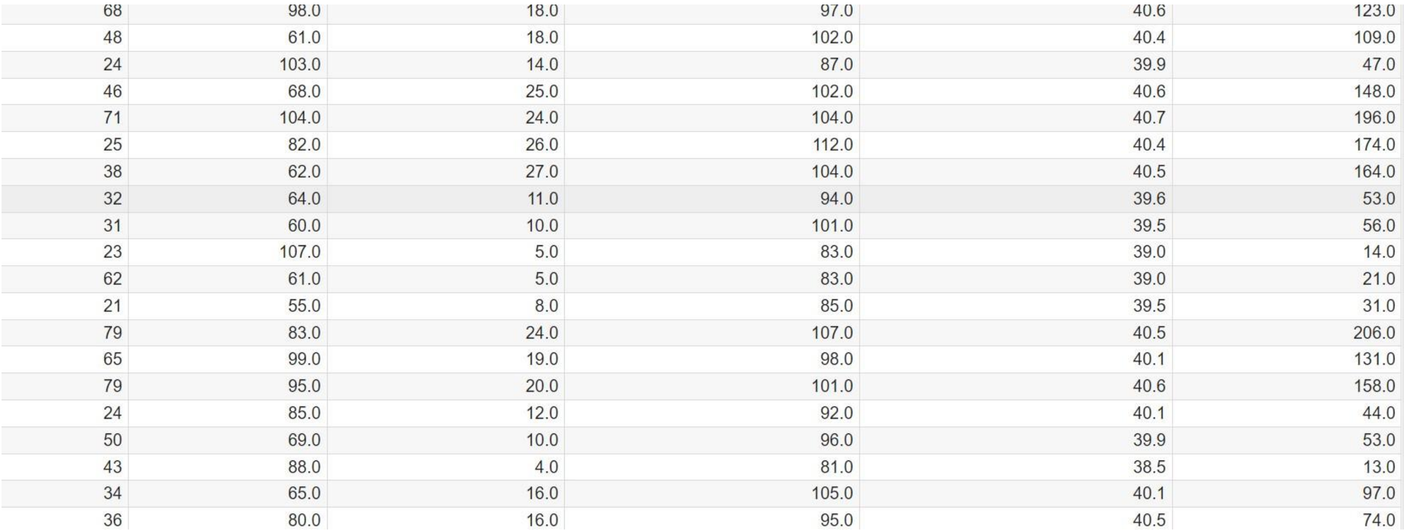
Age	Weight	Duration	Heart_Rate	Body_Temp	Calories
68	94.0	29.0	105.0	40.8	231.0
20	60.0	14.0	94.0	40.3	66.0
69	79.0	5.0	88.0	38.7	26.0
34	71.0	13.0	100.0	40.5	71.0
27	58.0	10.0	81.0	39.8	35.0
36	50.0	23.0	96.0	40.7	123.0
33	56.0	22.0	95.0	40.5	112.0
41	85.0	25.0	100.0	40.7	143.0
60	94.0	21.0	97.0	40.4	134.0
26	51.0	16.0	90.0	40.2	72.0
36	76.0	1.0	74.0	37.8	3.0
21	56.0	17.0	100.0	40.0	92.0
66	79.0	11.0	90.0	40.0	58.0
32	54.0	18.0	93.0	40.4	88.0
53	85.0	2.0	82.0	38.1	7.0
39	62.0	28.0	104.0	40.8	170.0
39	82.0	4.0	82.0	38.6	11.0
46	67.0	11.0	89.0	40.2	43.0

.

.

.

. .



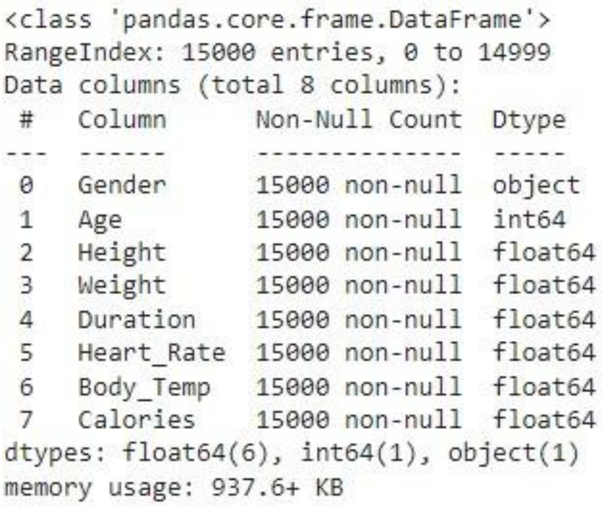
68	98.0	18.0	97.0	40.6	123.0
48	61.0	18.0	102.0	40.4	109.0
24	103.0	14.0	87.0	39.9	47.0
46	68.0	25.0	102.0	40.6	148.0
71	104.0	24.0	104.0	40.7	196.0
25	82.0	26.0	112.0	40.4	174.0
38	62.0	27.0	104.0	40.5	164.0
32	64.0	11.0	94.0	39.6	53.0
31	60.0	10.0	101.0	39.5	56.0
23	107.0	5.0	83.0	39.0	14.0
62	61.0	5.0	83.0	39.0	21.0
21	55.0	8.0	85.0	39.5	31.0
79	83.0	24.0	107.0	40.5	206.0
65	99.0	19.0	98.0	40.1	131.0
79	95.0	20.0	101.0	40.6	158.0
24	85.0	12.0	92.0	40.1	44.0
50	69.0	10.0	96.0	39.9	53.0
43	88.0	4.0	81.0	38.5	13.0
34	65.0	16.0	105.0	40.1	97.0
36	80.0	16.0	95.0	40.5	74.0

DATA PROCESSING

The data was processed to check if there were any missing values and outliers.

It is necessary to deal with the missing values and invalid outliers in the dataset as it can make the analysis results unreliable if the missing data is related to the response variable and if the outliers are influential.

```
#getting some information about the data
calories_data.info( )
```

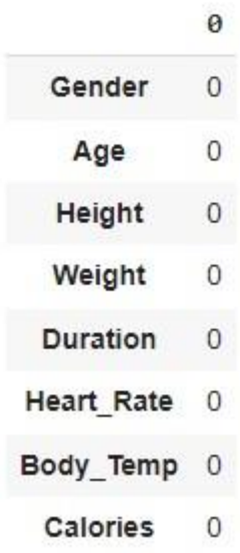


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Gender      15000 non-null  object 
 1   Age         15000 non-null  int64  
 2   Height      15000 non-null  float64
 3   Weight      15000 non-null  float64
 4   Duration    15000 non-null  float64
 5   Heart_Rate  15000 non-null  float64
 6   Body_Temp   15000 non-null  float64
 7   Calories    15000 non-null  float64
dtypes: float64(6), int64(1), object(1)
memory usage: 937.6+ KB
```

Here, we can identify the data type of each variable. Gender is not numerical, so we may not consider it for our further analysis. In this project we are going to deal with quantitative variables only as we are going to fit Multiple Linear Regression Model on our data.

All the columns are not null.

```
# checking for missing values
calories_data.isnull().sum()
```

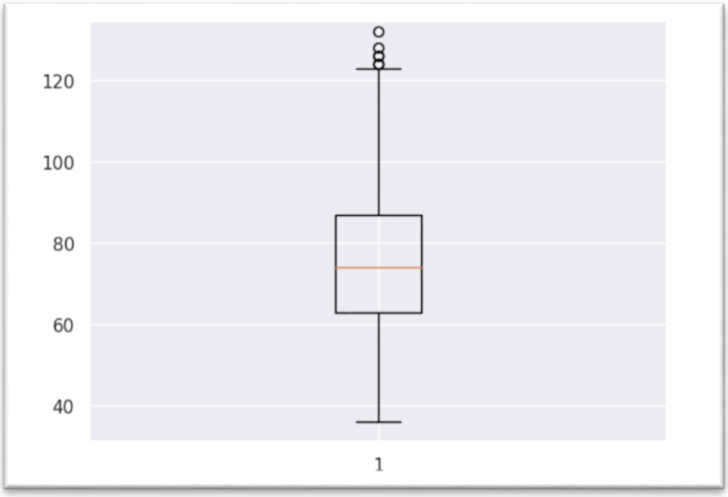


	0
Gender	0
Age	0
Height	0
Weight	0
Duration	0
Heart_Rate	0
Body_Temp	0
Calories	0

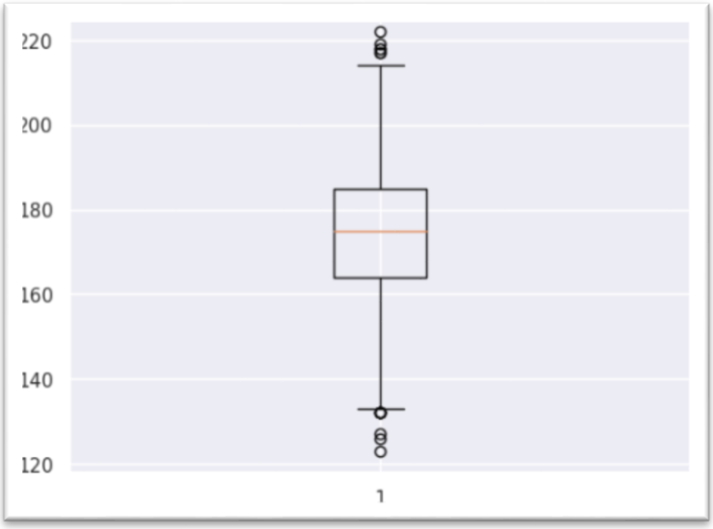
There are no missing values in our data.

An outlier is defined as an observation point that is distant from the main screen value. One of the most efficient ways to identify outliers is Data visualisation: Scatter plot, Box Plot or Histogram are the methods to identify them. Here, I have used Box Plots.

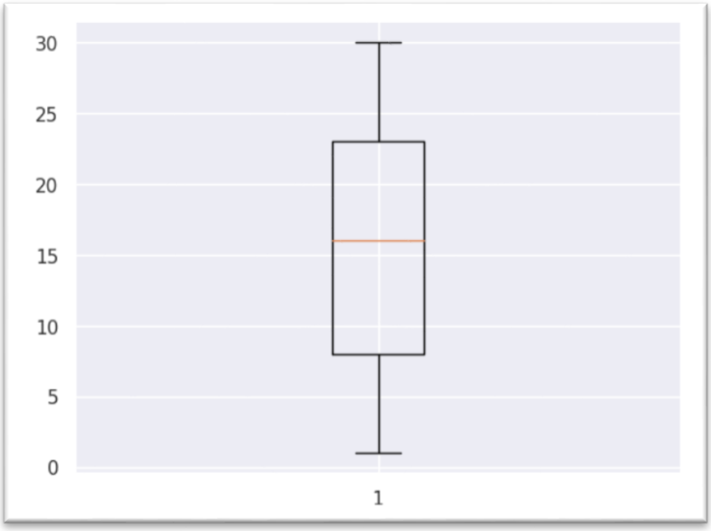
```
#Boxplot to visualise outliers plt.boxplot
(calories_data['Weight'])
```



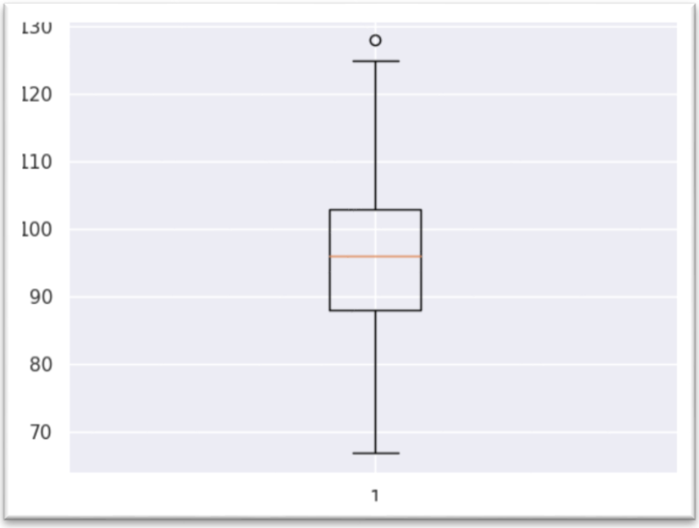
```
plt.boxplot (calories_data['Height'])
```



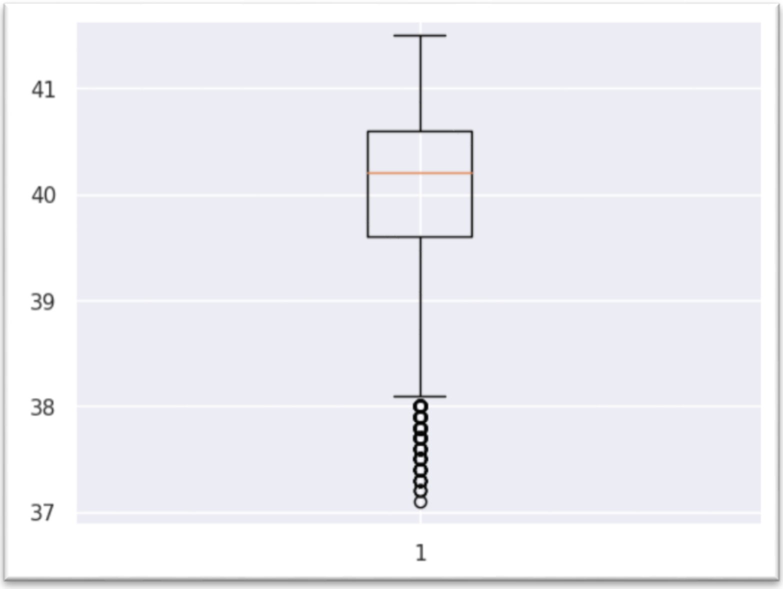
```
plt.boxplot (calories_data ['Duration'])
```



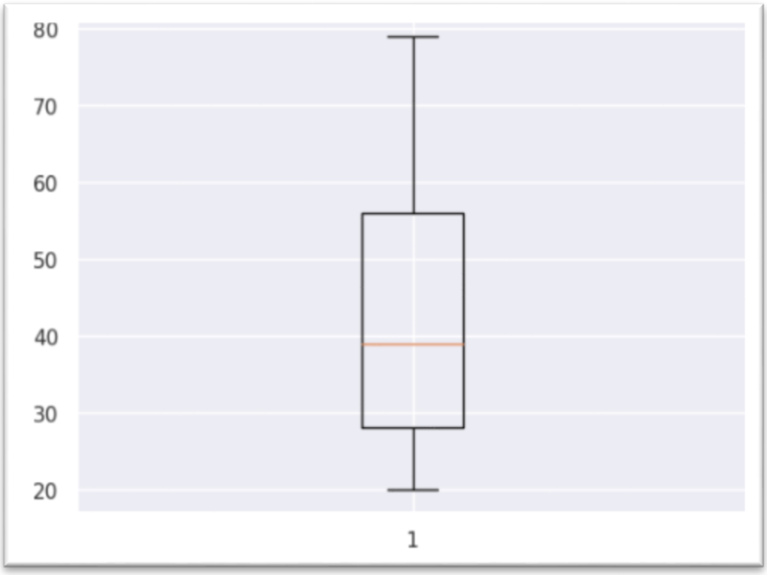
```
plt.boxplot (calories_data ['Heart_Rate'])
```



```
plt.boxplot (calories_data ['Body_Temp'])
```



```
plt.boxplot (calories_data ['Age'])
```



Outliers are sometimes bad values which can be due to faulty measurements or incorrect recording. These can be discarded from the data, but there must be a strong non-statistical evidence to delete them.

In our data, outliers represent natural variations in the population, and they should be left as is in the dataset. As sometimes outliers are unusual but **plausible** observations.

We have found outliers in variable ‘Weight’ which is more than 120kg. There are people who are overweight and burn calories by doing some exercises, there is no point of discarding these values. Similarly for ‘Height’ there are outliers below and above the minimum and maximum value point. It cannot be discarded as it doesn’t seem to be faulty measurements.

There are no outliers in the variable ‘Duration’ and ‘Age’.

There is only one distant point in the variable ‘Heart rate’ which is more than others points, It can be due to intense physical activities.

The box plot of ‘Body Temperature’ shows distant points below the Minimum value. This can be due to people performing low intensity physical activities like walking which doesn’t affects the body temperature much and it stays normal 37-38 degree Celsius.

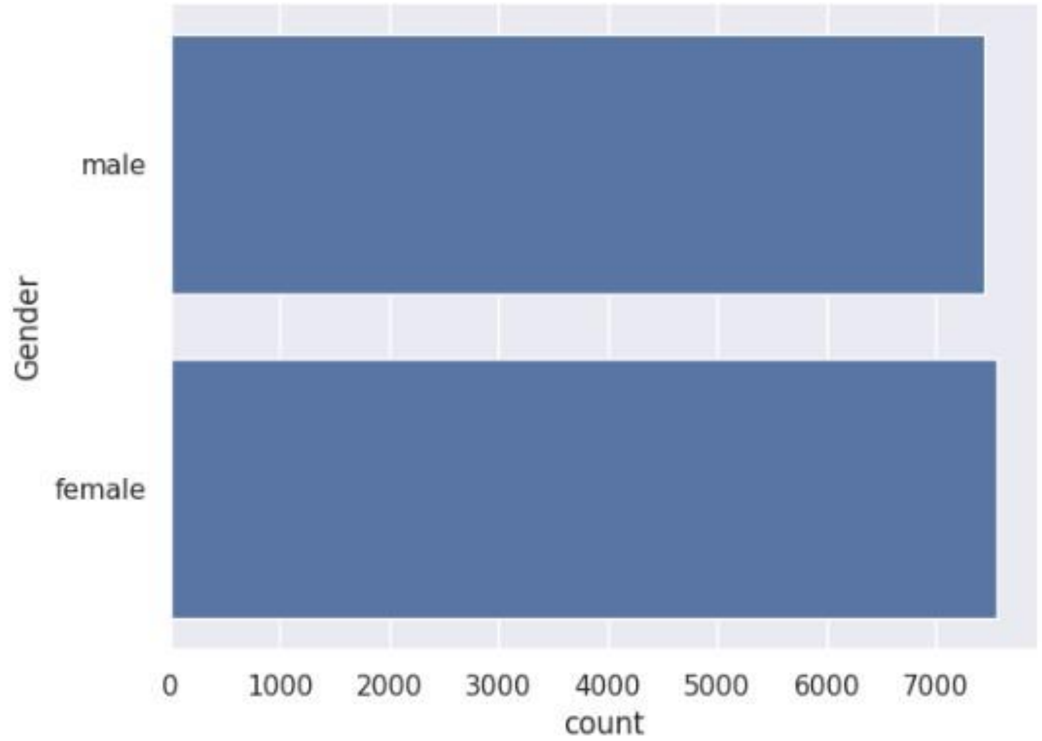
As we did not find any valid reason to prove that the outliers are bad values, therefore we don’t discard any of the values since they all have natural variations.

DESCRIPTIVE STATISTICS

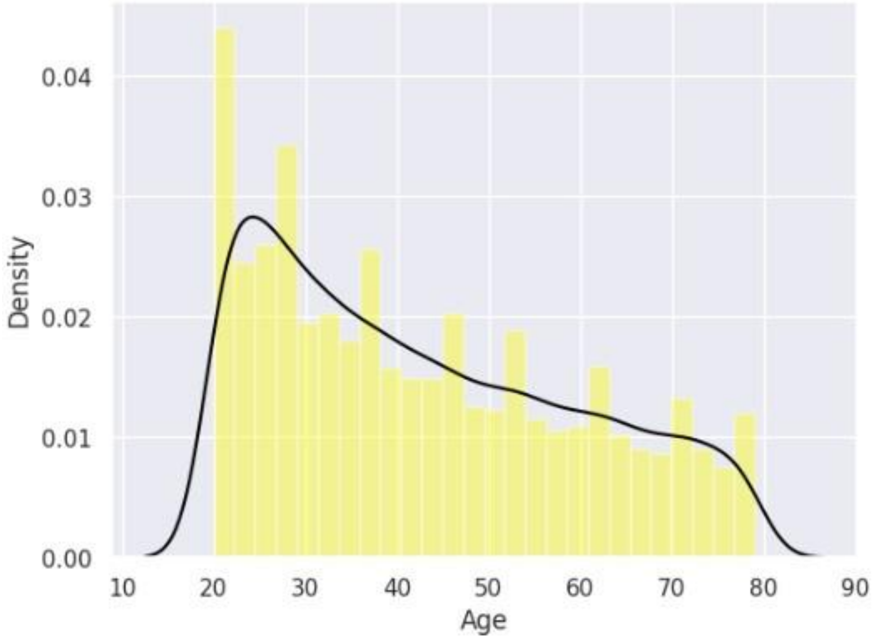
1. Visualizing the Data set

```
sns.countplot (calories_data ['Gender'])
```

```
<Axes: xlabel='count', ylabel='Gender'>
```

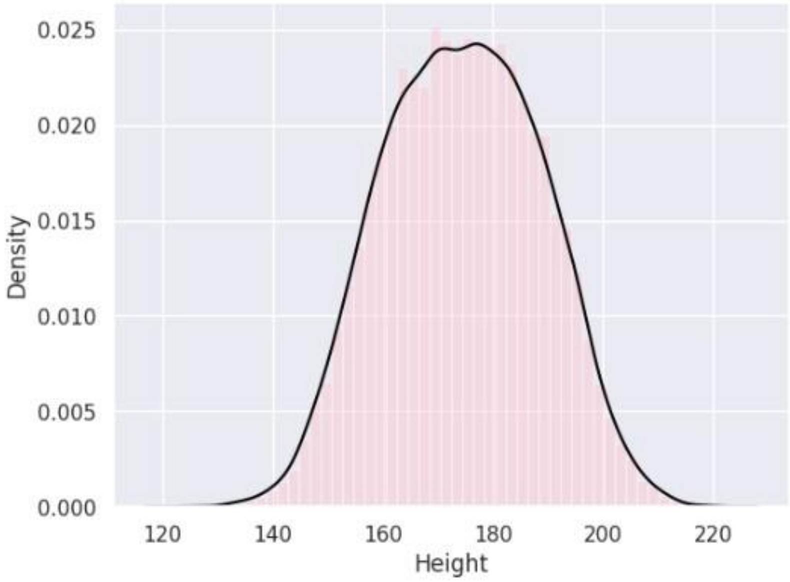


```
sns.distplot (calories_data ['Age'], hist_kws= {'color': 'yellow'}, kde_kws= {'color': 'black'})
```



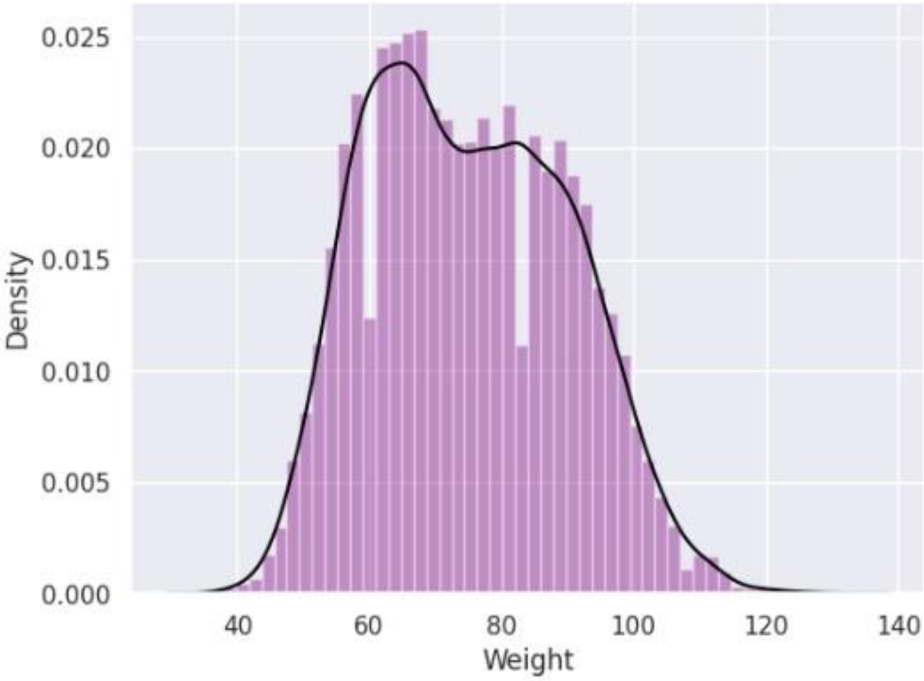
- Most of the people in the age group 20-80 years perform physical activities. This graph depicts positively skewed distribution of the variable ‘Age’. It implies that as age increases number of people engaging in physical activities decreases.

```
sns.distplot (calories_data ['Height'], hist_kws= {'color': 'pink'}, kde_kws= {'color': 'black'})
```



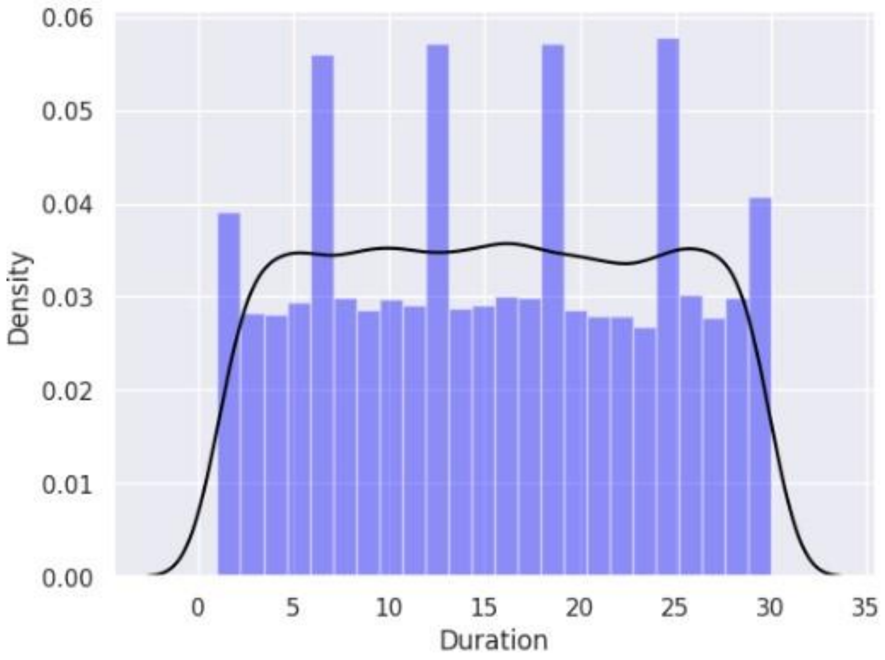
- The graph of Height represents a normal distribution. It implies Maximum people have height ranging between 160-180 cm.

```
sns.distplot (calories_data ['Weight'], hist_kws= {'color': 'purple'}, kde_kws= {'color': 'black'})
```



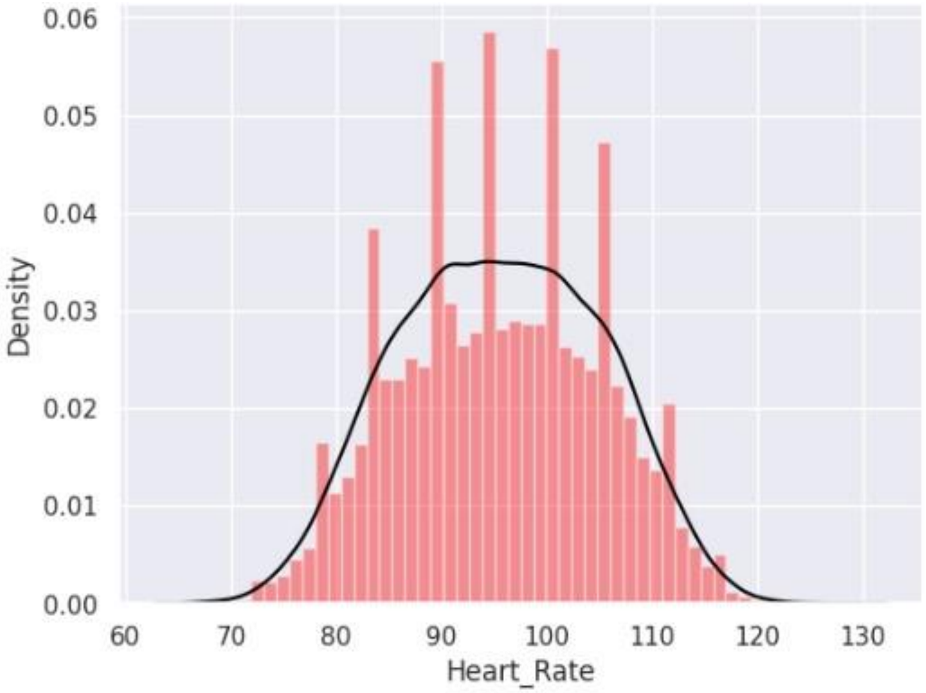
- Maximum people around age 70 kg engage in workouts.

```
(calories_data ['Duration'], hist_kws= {'color': 'blue'}, kde_kws= {'color': 'black'})
```

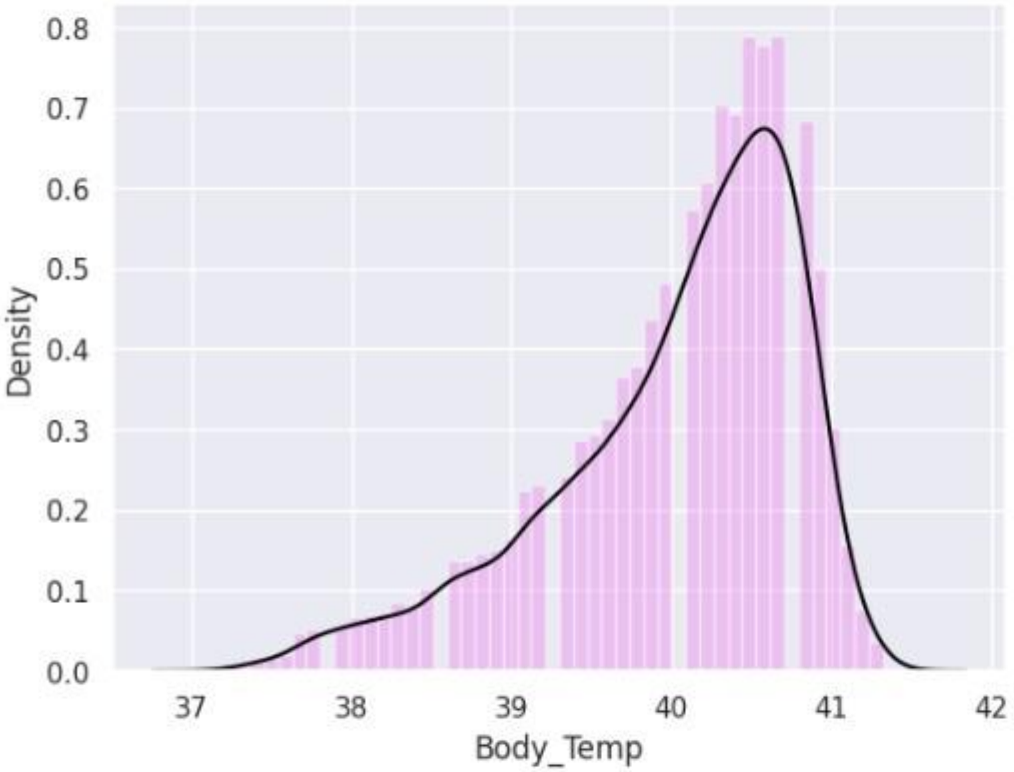


- ➢ Duration graph depicts Uniform distribution roughly, that is duration of doing workouts ranges between 0 to 30 minutes.

```
sns.distplot (calories_data ['Heart_Rate'], hist_kws= {'color': 'red'}, kde_kws= {'color': 'black'})
```



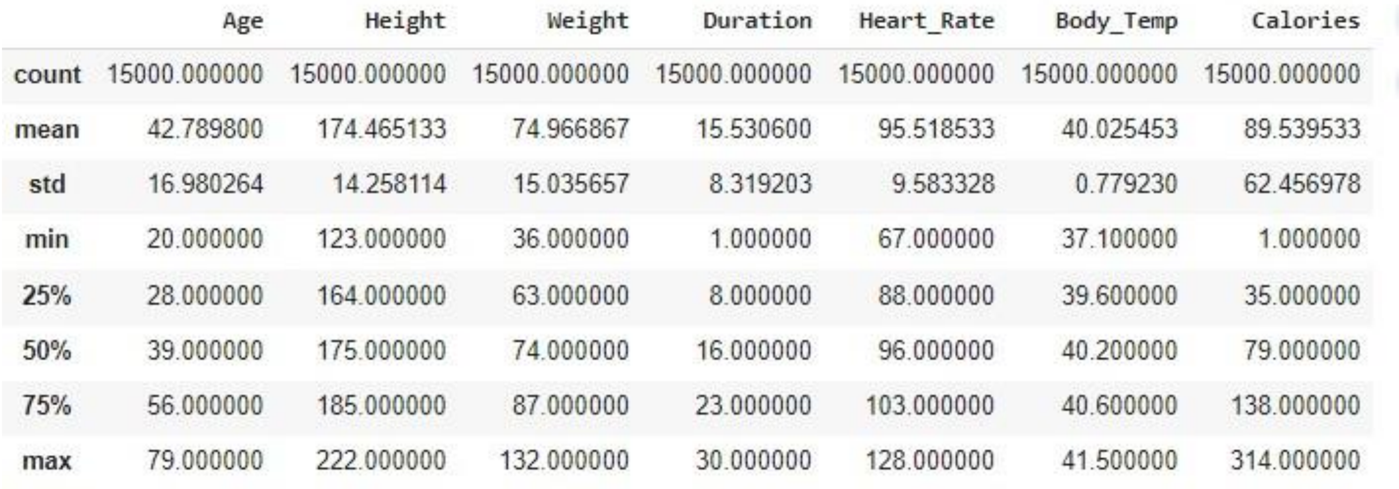
- ➢ Heart rate graph represents- Most of the people in the data perform medium intensity workouts. sns.distplot (calories_data ['Body_Temp'], hist_kws= {'color': 'violet'}, kde_kws= {'color': 'black'})



- Body Temperature graph is skewed negatively, it implies most of the people in the data have higher body temperature than the normal temperature 37 degree Celsius.

2. Statistical Measures

```
#getting some statistical measures about the data calories_data.describe
()
```

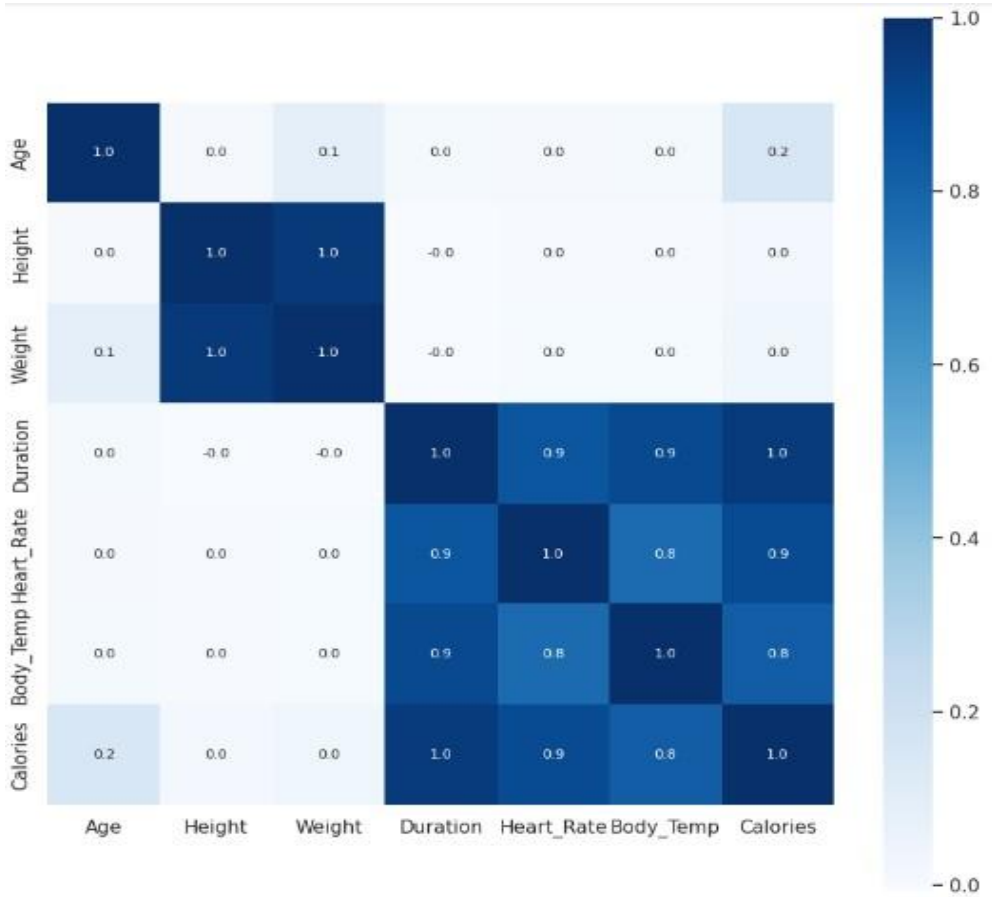


	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
count	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000
mean	42.789800	174.465133	74.966867	15.530600	95.518533	40.025453	89.539533
std	16.980264	14.258114	15.035657	8.319203	9.583328	0.779230	62.456978
min	20.000000	123.000000	36.000000	1.000000	67.000000	37.100000	1.000000
25%	28.000000	164.000000	63.000000	8.000000	88.000000	39.600000	35.000000
50%	39.000000	175.000000	74.000000	16.000000	96.000000	40.200000	79.000000
75%	56.000000	185.000000	87.000000	23.000000	103.000000	40.600000	138.000000
max	79.000000	222.000000	132.000000	30.000000	128.000000	41.500000	314.000000

3. Correlation between the variables in the dataset #constructing

a heatmap to understand the correlation

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



- Duration and Calories are perfectly positively correlated. It implies that: Longer workouts typically burn more calories.
- Heart rate and calories are strongly positively correlated. I.e. As duration and intensity of workout increase the heart rate so heart rate increased leads to burning more calories.
- Body temperature and calories are strongly positively correlated. Body generates heat when working harder burning more calories.
- In this dataset, height and weight do not directly impact on calories burned.
- Height and weight are perfectly positively correlated.
- We can conclude that Duration plays a more significant role in burning calories, which is also true according to the theory.

- *Here we see that Height and Weight are perfectly positively correlated. ○ In Regression Analysis, if any two variables are perfectly correlated (+1/-1), it means that Multicollinearity is likely to be a problem in the regression analysis.*
- *Hence, to avoid the problem of Multicollinearity, we may drop one of the two variables Height or Weight or we could fit Ridge Regression.*
- *Simple least squares regression needs that the predictor variables are independent. We could tolerate small correlations but the problem gets serious if the variables are perfectly collinear. It's common to drop some of the correlated variables, keeping the most meaningful.*
- *According to the theory, Weight is more related to the number of calories burned during a workout than height. Hence we will drop Height so as to reduce Multicollinearity.*

ANALYSIS

1. DEFINING VARIABLES

Dependent Variable Y: Calories Burnt during exercise
Independent Variables are
X1: Age of the person

X2: Weight of the person
X3: Duration of exercise
X4: Heart Rate of the person
X5: Body Temperature of the person

2. MODEL

Multiple Linear Regression Model for k explanatory variables is given by,

$Y = \beta 0 + \beta 1\ X1 + \beta 2\ X2 + \ldots\ \beta k\ Xi + \varepsilon i$
Where
Y: Dependent Variable $\beta 0$:
Intercept term
$\beta 1, \beta 2, \ldots, \beta i$: Regression Coefficients
X1, X2,..., Xi: Regressors εi:
Error term

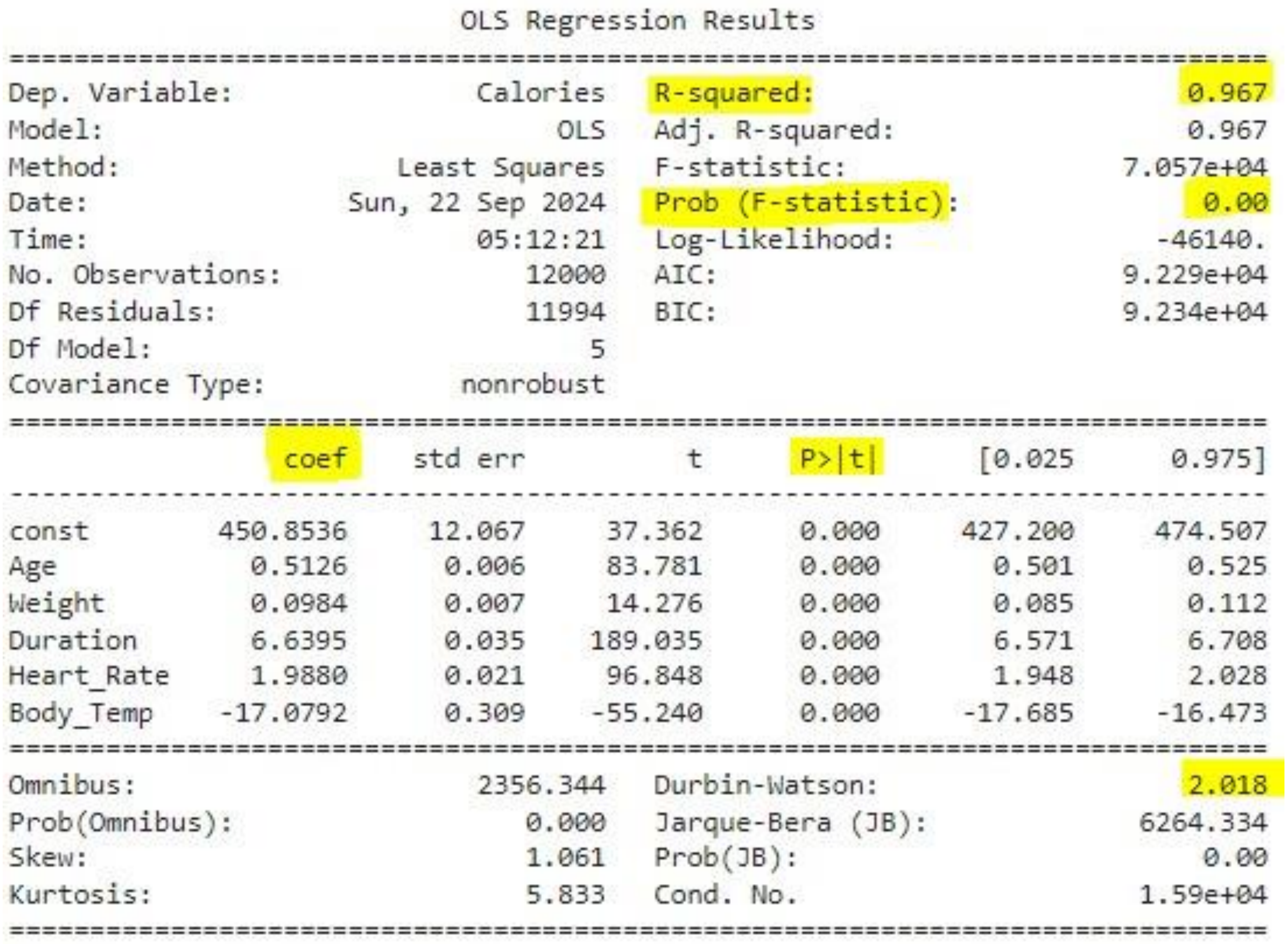
For our analysis, we have 5 Explanatory Variables.
Therefore our Model will be,
$Y = \beta 0 + \beta 1\ X1 + \beta 2\ X2 + \beta 3\ X3 + \beta 4\ X4 + \beta 5\ X5 + \varepsilon i$ (i=6)

Fitted Model is
$\hat{Y} = \hat{\beta} 0 + \hat{\beta} 1\ X1 + \hat{\beta} 2\ X2 + \hat{\beta} 3\ X3 + \hat{\beta} 4\ X4 + \hat{\beta} 5\ X5$

3. REGRESSION

- As we are building the regression model using machine learning, we split the data into a training set (80% of the data) and a test set (20% of the data) for model training and evaluation.
- In a basic two-part data split, the training data set is used to train and develop models. Training sets are commonly used to estimate different parameters or to compare different model performance. The testing data set is used after the training is done.

```
import statsmodels.api as sm
# Add constant term to independent variables
X_train_const = sm.add_constant (X_train)
X_test_const = sm.add_constant (X_test)
# fit the model
model = sm.OLS (Y_train, X_train_const).fit ()
# Print the summary of the model print
(model.summary ())
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Calories   R-squared:                       0.967
Model:                            OLS   Adj. R-squared:                  0.967
Method:                 Least Squares   F-statistic:                 7.057e+04
Date:                Sun, 22 Sep 2024   Prob (F-statistic):               0.00
Time:                        05:12:21   Log-Likelihood:                -46140.
No. Observations:               12000   AIC:                         9.229e+04
Df Residuals:                   11994   BIC:                         9.234e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        450.8536     12.067     37.362      0.000     427.200     474.507
Age            0.5126      0.006     83.781      0.000       0.501       0.525
Weight         0.0984      0.007     14.276      0.000       0.085       0.112
Duration       6.6395      0.035    189.035      0.000       6.571       6.708
Heart_Rate     1.9880      0.021     96.848      0.000       1.948       2.028
Body_Temp    -17.0792      0.309    -55.240      0.000     -17.685     -16.473
==============================================================================
Omnibus:                     2356.344   Durbin-Watson:                   2.018
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6264.334
Skew:                           1.061   Prob(JB):                         0.00
Kurtosis:                       5.833   Cond. No.                     1.59e+04
==============================================================================
```

CONCLUSIONS FROM THE REGRESSION RESULTS

- **Coefficient of Determination (R^2):** R^2 is a goodness of fit measure for linear regression models.
- **Adjusted R^2:** Corrected goodness of fit (model accuracy) measures for linear models **R^2 = 0.967, Adjusted R^2 = 0.967**

It implies that 96.7% of variation in Y (calories burnt) is explained by the independent variables.

- **The fitted linear regression model for our data is**

$\hat{Y}$= 450.8536 + 0.5126 (Age) + 0.0984 (Weight) + 6.6395 (Duration) + 1.9880 (Heart_Rate) – 17.0792 (Body_Temp)

1) As age increases by 1 year, calories burnt increases by 0.5 cal.
2) As weight increases by 1 kg, calories burnt increases by 0.0984 cal.
3) As duration of workout increases by 1 minute, calories burnt increases by 6.6395 cal.
4) As heart rate increases by 1 bpm, calories burnt increases by 1.9880 cal.
5) As body temperature increases by 1 degree Celsius, calories burnt decreases by 17.0792 cal. [Being cold burns more calories than being hot because the body works harder to maintain a balanced temperature. However, the rate of metabolism increases as temperature increases, and then declines rapidly at higher temperatures]

4. OVERALL SIGNIFICANCE

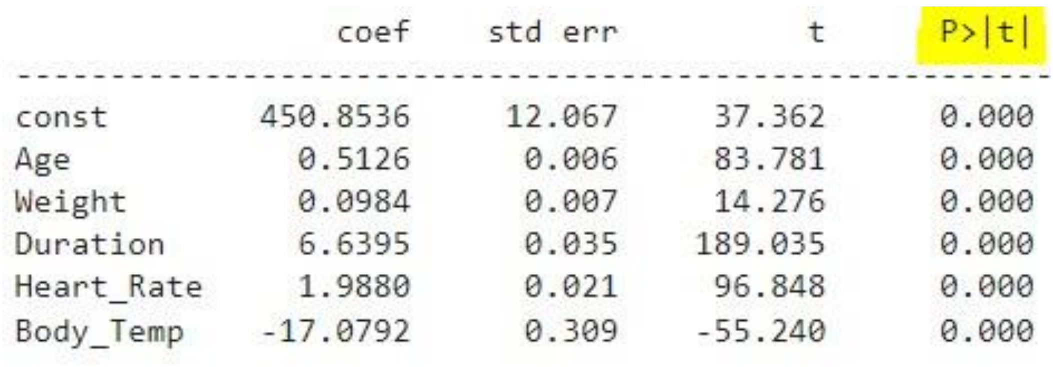
```
print(model.fvalue)
print(model.f_pvalue)
```

Fcal = 70565.7088463788 p-value
= 0.0
(p-value is calculated under the assumption that the Ho is true, which means the model is insignificant)

Hypothesis to be tested:
Ho: β1= β2 = β3 = β4 = β5 = β6 = 0 (ie. Model is not significant.) H1:
at least one βi ≠ 0 (i.e. Model is significant.)

Since pvalue is 0 which is less than Level of significance (0.05)
(Level of significance represents the level of evidence required to reject the Null Hypothesis, Ho) Therefore, We Reject Ho at 5% L.O.S.
Model is significant

5. INDIVIDUAL SIGNIFICANCE

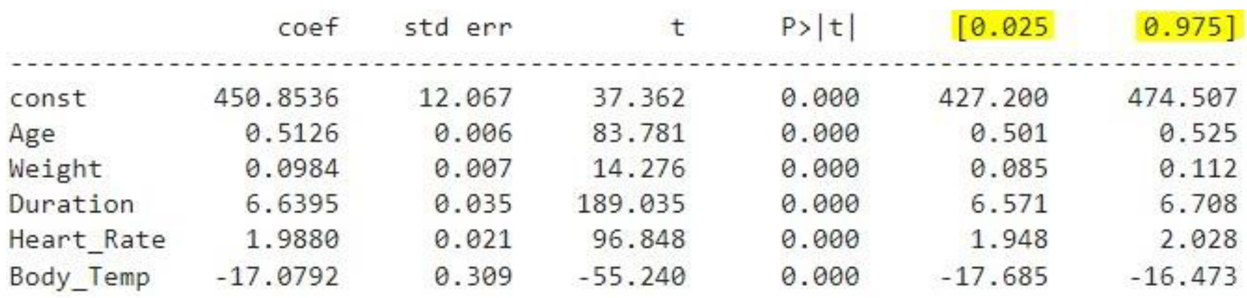


```
                coef    std err          t      P>|t|
-----------------------------------------------------
const       450.8536     12.067     37.362      0.000
Age           0.5126      0.006     83.781      0.000
Weight        0.0984      0.007     14.276      0.000
Duration      6.6395      0.035    189.035      0.000
Heart_Rate    1.9880      0.021     96.848      0.000
Body_Temp   -17.0792      0.309    -55.240      0.000
```

From the OLS Regression Results, we see that all the p-values for the Regressors is 0 which is < 0.05 (L.O.S.)

This implies that **all the Regressors are Significant.**

6. 95% CONFIDENCE INTERVAL FOR THE REGRESSION COEFFICIENTS

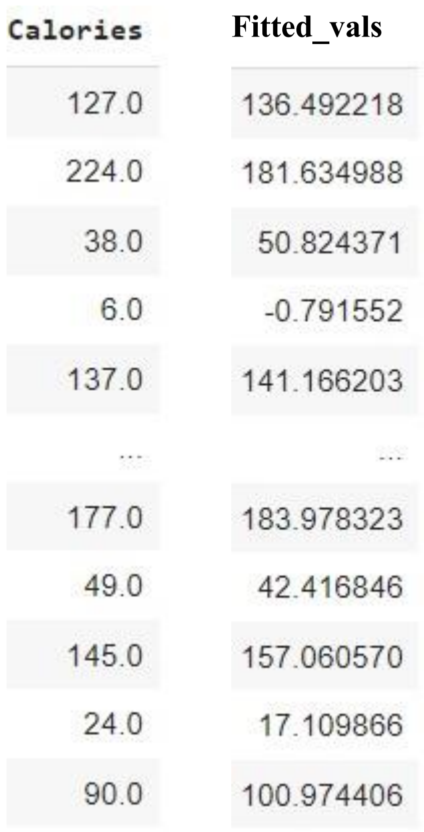


```
                coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       450.8536     12.067     37.362      0.000     427.200     474.507
Age           0.5126      0.006     83.781      0.000       0.501       0.525
Weight        0.0984      0.007     14.276      0.000       0.085       0.112
Duration      6.6395      0.035    189.035      0.000       6.571       6.708
Heart_Rate    1.9880      0.021     96.848      0.000       1.948       2.028
Body_Temp   -17.0792      0.309    -55.240      0.000     -17.685     -16.473
```

- The 95% CI for the regression coefficients are shown above, it is a range of values that is likely to contain the true mean of a population.
- It can be interpreted as there is 95% assurance of the regression coefficients falling in their respective intervals for the population.

7. FITTED VALUES (PREDICTED VALUES)

```
y_pred = model.predict (X_test_const)
print (y_pred)
```



Calories	Fitted_vals
127.0	136.492218
224.0	181.634988
38.0	50.824371
6.0	-0.791552
137.0	141.166203
...	...
177.0	183.978323
49.0	42.416846
145.0	157.060570
24.0	17.109866
90.0	100.974406

8. MEAN ABSOLUTE ERROR

```
#obtain mae from sklearn.metrics import
mean_absolute_error mae =
mean_absolute_error(Y_test, y_pred) print(mae)
```

8.395852655123567
Is the mean absolute error.

Mean Absolute Error (MAE) is a measure of the average size of the mistakes in a collection of predictions, without taking their direction into account. It is measured as the average absolute difference between the predicted values and the actual values and is used to assess the effectiveness of a regression model.

After fitting of model for our data using OLS Regression technique,
We get an average difference between predicted and true values as 8 calories.

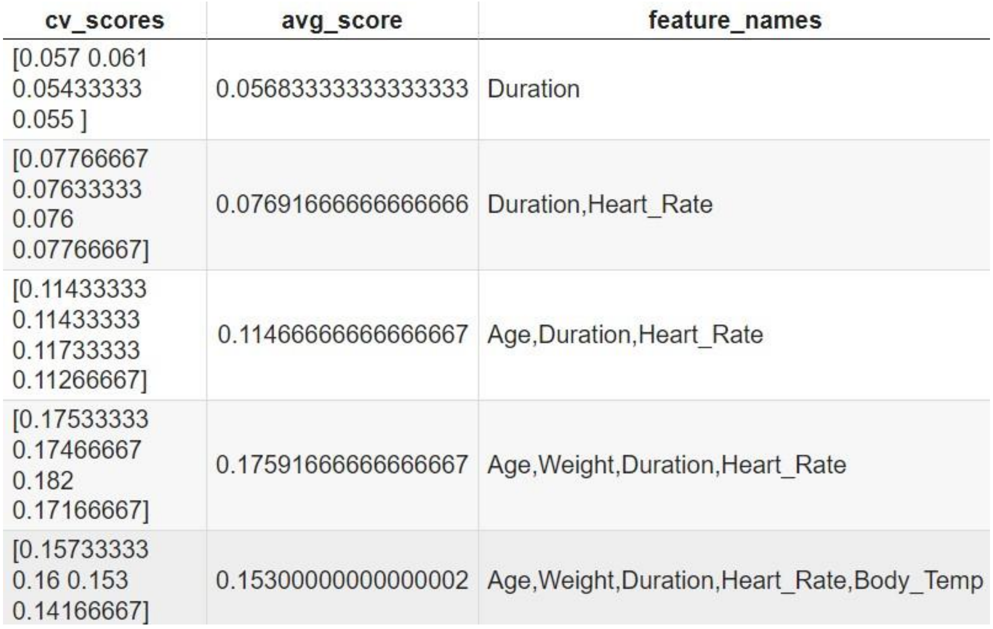
Since, the data range is large, an MAE of 8 might be expectable.

VARIABLE SELECTION

Variable Selection method refers to the process of choosing the most relevant variables to include in a regression model.
We will be using **Forward Selection method** which is a technique that starts with an empty set of features and adds the most predictive feature in each iteration until a stopping criterion is met.

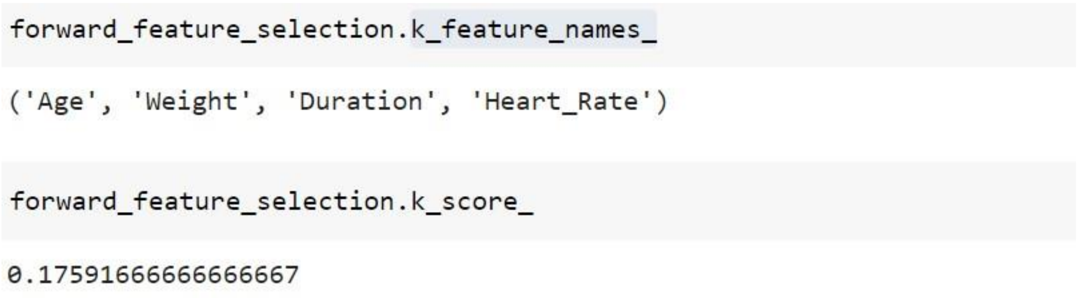
```
forward_feature_selection = SequentialFeatureSelector(RandomForestClassifier(n_jobs=1),




                                             k_features=(1,5),
floating=False,
forward=True,                                                   verbose=2,
scoring='accuracy',
cv=4).fit(X_train,Y_train)
```

```
pd.DataFrame.from_dict(forward_feature_selection.get_metric_dict()).T
```



cv_scores	avg_score	feature_names
[0.057 0.061 0.05433333 0.055]	0.05683333333333333	Duration
[0.07766667 0.07633333 0.076 0.07766667]	0.07691666666666666	Duration,Heart_Rate
[0.11433333 0.11433333 0.11733333 0.11266667]	0.11466666666666667	Age,Duration,Heart_Rate
[0.17533333 0.17466667 0.182 0.17166667]	0.17591666666666667	Age,Weight,Duration,Heart_Rate
[0.15733333 0.16 0.153 0.14166667]	0.15300000000000002	Age,Weight,Duration,Heart_Rate,Body_Temp

According to the algorithm of forward variable selection method,
Forward selection starts with no variables in the model and adds variables one at a time until no variable can significantly improve the model.
We can see that the average score improves on adding the independent variables one by one.
The average score is the highest when the model takes the variables Age, Weight, Duration and Heart Rate.
As it adds the variable Body temperature, the average score reduces and thus not improving the model.
Therefore, the selected variables are



```
forward_feature_selection.k_feature_names_
```

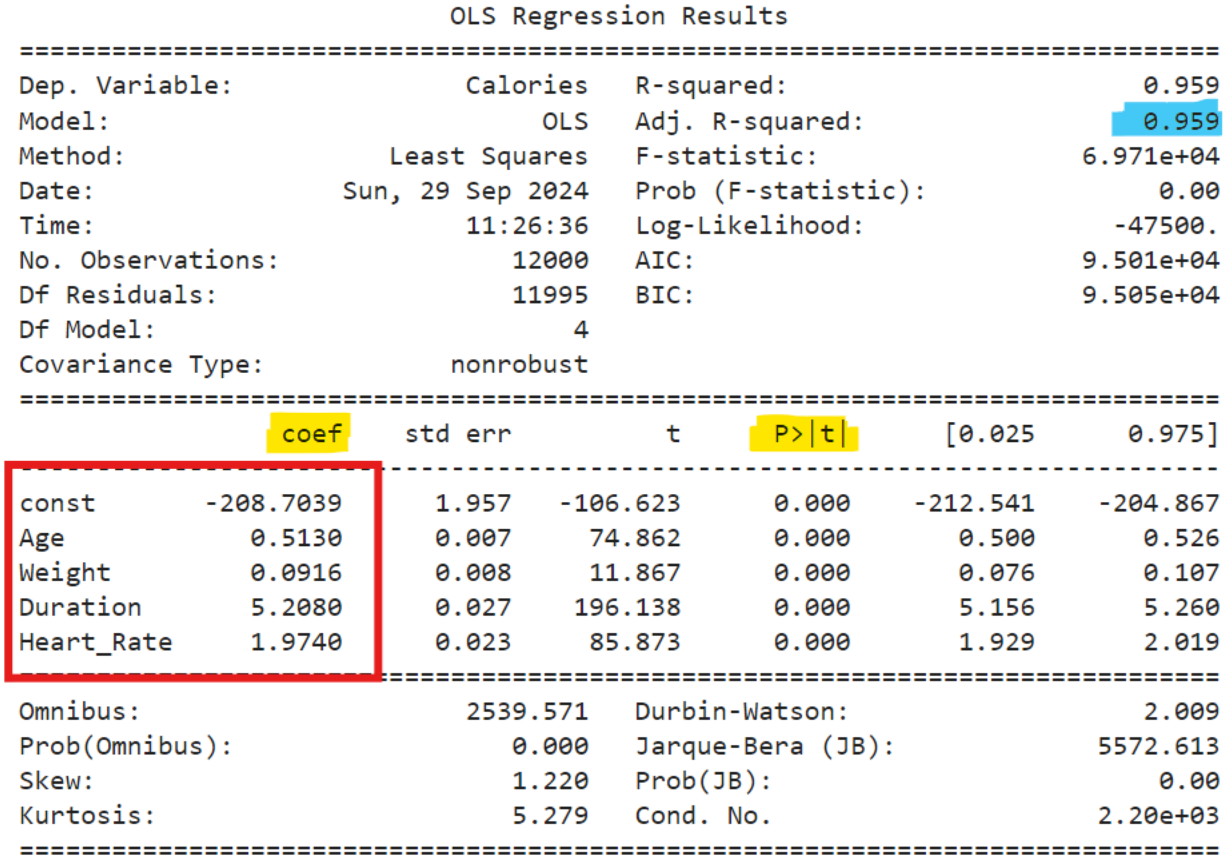
```
('Age', 'Weight', 'Duration', 'Heart_Rate')
```

```
forward_feature_selection.k_score_
```

```
0.17591666666666667
```

After dropping the variable Body_Temp from our model.

We get the following results:



```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:               Calories   R-squared:                       0.959
Model:                            OLS   Adj. R-squared:                  0.959
Method:                 Least Squares   F-statistic:                 6.971e+04
Date:                Sun, 29 Sep 2024   Prob (F-statistic):               0.00
Time:                        11:26:36   Log-Likelihood:                -47500.
No. Observations:               12000   AIC:                         9.501e+04
Df Residuals:                   11995   BIC:                         9.505e+04
Df Model:                           4                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       -208.7039      1.957   -106.623      0.000    -212.541    -204.867
Age            0.5130      0.007     74.862      0.000       0.500       0.526
Weight         0.0916      0.008     11.867      0.000       0.076       0.107
Duration       5.2080      0.027    196.138      0.000       5.156       5.260
Heart_Rate     1.9740      0.023     85.873      0.000       1.929       2.019
==============================================================================
Omnibus:                     2539.571   Durbin-Watson:                   2.009
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             5572.613
Skew:                           1.220   Prob(JB):                         0.00
Kurtosis:                       5.279   Cond. No.                     2.20e+03
==============================================================================
```

Therefore
Our final model becomes

$$\hat{Y} = -208.7039 + 0.5130\ (Age) + 0.0916\ (Weight) + 5.2080\ (Duration) + 1.9740\ (Heart\_Rate)$$

With 4 regressors: Age, Weight, Duration and Heart_Rate

Here, the coefficient term is insignificant so we may ignore it.

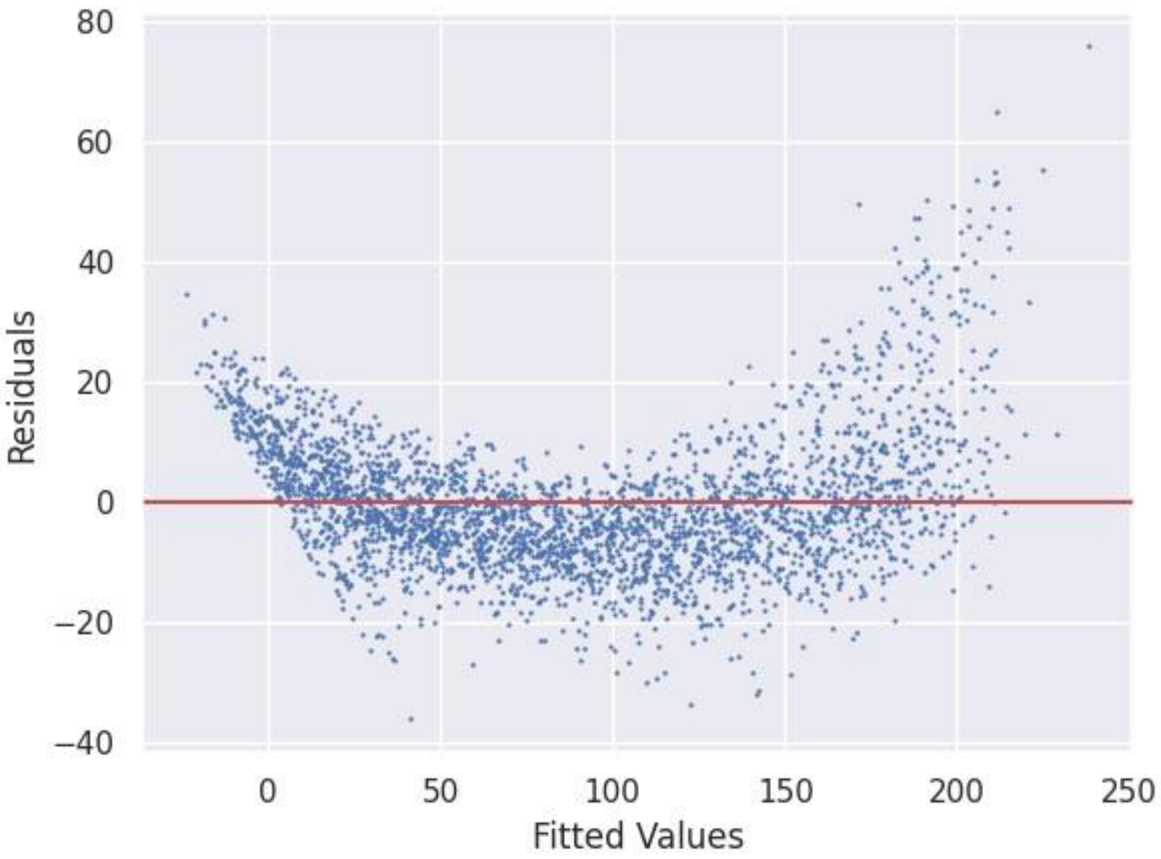
RESIDUAL ANALYSIS

Residual is defined as the difference between the observed and the fitted values of the study variable. Residual can be viewed as the deviation between the data and the fit. So, it is also a measure of the variability in the response variable that is not explained by the regression model.
If there is any departure from the assumptions on random errors, then it can be shown up by the residual. The analysis of residuals helps in finding the model inadequacies.

The graphical analysis of residuals is a very effective way to investigate the adequacy of the fit of a regression model and to check the underlying assumptions.

Plot of residuals against the fitted values:

```
plt.scatter (y_pred, residuals, s=0.5)
plt.axhline (y = 0, color = 'r')
plt.xlabel ('Fitted Values') plt.ylabel
('Residuals') plt.show ()
```



- ➢ The residuals are contained inside a curved plot, it indicates non-linearity.

- ➢ **The assumed relationship between Y and X's is *non-linear.***

CHECKING OF ASSUMPTIONS

1. AUTOCORRELATION

```
#Test of Autocorrelation
from statsmodels.stats.stattools import durbin_watson

print ('d =', durbin_watson (residuals))
```

d = 2.009

Durbin-Watson value is close to 2 **Therefore there is no autocorrelation.**

2. HETEROSCEDESTICITY

```
# Test of Heteroscedesticity #perform
Breusch-Pagan test
from statsmodels.stats.diagnostic import het_breuschpagan

name = ['Lagrange multiplier statistic', 'p-value', 'f-value', 'f p-
value'] test = het_breuschpagan(model.resid, X_train_const)
print(name) print(test)
```

```
['Lagrange multiplier statistic', 'p-value', 'f-value', 'f p-value']
(474.72671187854274, 1.9572387289926205e-101, 123.51869596992569, 1.7410099104002112e-103)
```

Ho: Heteroscedasticity is not present. vs. H1: Heteroscedasticity is present.

- Lagrange Multiplier (LM) Statistic: 944.27

- This measures the magnitude of Heteroscedasticity.
- A large value indicates strong evidence of Heteroscedasticity.

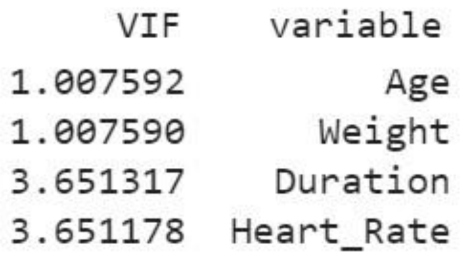
- p-value: 6.98e-202

- Extremely small p-value (< 0.05) indicates strong evidence against homoscedasticity. We reject Ho at 5% LOS.

Heteroscedasticity is present in the residuals.

3. MULTICOLLINEARITY

```
# Test of Multicollinearity
from statsmodels.stats.outliers_influence import variance_inflation_factor

vif = pd.DataFrame()
vif['VIF'] = [variance_inflation_factor(X_train_const.values, i) for i in range(X_train_const.shape[1])]
vif ['variable'] = X_train_const.columns print(vif)
```



```
     VIF    variable
1.007592         Age
1.007590      Weight
3.651317    Duration
3.651178  Heart_Rate
```

A VIF value of less than 5 indicates that Multicollinearity is not present.

INFERENCE:

- After checking for the assumptions we found that the residuals indicate non-linearity and so linear regression model does not fit on this data for calories burnt prediction.
- Also, there is a presence of heteroscedasticity in the residuals.
- Therefore, we fit a machine learning model, XGBoost Regression on our data as it can be very powerful at modelling non-linear and complex relationships
- Although the linear regression model provided few valuable insights into the relationship between the predictors and calories burnt, we further explore the data using XGBoost regression to potentially improve model performance and capture non-linear relationships.

Extreme Gradient Boosting (XGBoost) Regression

XGBoost Regression is a supervised learning algorithm that predicts continuous outcomes by combining multiple weak models. It handles non-linear relationships, is robust to outliers, and processes in parallel, making it efficient. XGBoost initializes a base model, computes residuals, creates new decision trees, and updates the model iteratively. This results in high accuracy, flexibility, and speed. Ideal for regression tasks, time series forecasting, and recommendation systems, XGBoost Regression is a powerful tool for predictive modeling, offering superior performance and interpretability.

```
model= XGBRegressor()
model.fit(X_train, Y_train)

metrics.mean_absolute_error(Y_test, test_data_prediction) metrics.r2_score(Y_test,
test_data_prediction)
```

R^2= 0.9951385074975367

99.5% variation in Y is explained by the regressors.

```
metrics.mean_absolute_error(Y_test, test_data_prediction)
```

Mean absolute error = 2.4666451590756577

The average difference between predicted and actual values is approximately 2.47 units.

- **With this regression model , we can predict calories burned more accurately using machine learning techniques.**

CONCLUSION

In this study, we aimed to predict calories burned using various independent variables such as age, weight, duration of exercise, heart rate, and body temperature.

Initially, a Multiple Linear Regression (MLR) model was used to analyze the data. The model performed well, explaining 96.7% of the variation in calories burned, as indicated by the R^2 value. We used the Forward Selection method to choose the most predictive variables. This process led to the exclusion of body temperature, as its inclusion reduced the model's performance. The final model included age, weight, duration, and heart rate as the key predictors, significantly improving the model’s accuracy. However, residual analysis revealed non-linearity, and heteroscedasticity was also present, making the linear model less suitable for this data.

To address these issues, we applied an XGBoost regression model, which is more adept at handling nonlinear relationships. The XGBoost model improved the prediction accuracy significantly, with an R^2 of 99.5% and a lower Mean Absolute Error,MAE of 2.47 compared to the linear regression model's MAE of 8.

In conclusion, while the linear regression model provided a foundational understanding of the relationships between variables, it did not fully satisfy the underlying assumptions. The XGBoost model, on the other hand, offered a more accurate and reliable prediction, demonstrating that advanced machine learning techniques are better suited for datasets with non-linearity and heteroscedasticity.

REFERENCES

https://www.geeksforgeeks.org/calories-burnt-prediction-using-machine-learning/

https://www.researchgate.net/publication/375147744_Calories_Burnt_Prediction_Using_Machine_Learning_Approach

https://www.itm-conferences.org/articles/itmconf/pdf/2023/04/itmconf_I3cs2023_01010.pdf

https://www.geeksforgeeks.org/ml-xgboost-extreme-gradient-boosting/

Source of data https://www.kaggle.com/datasets/fmendes/fmendesdat263xdemos