Calories Burnt Prediction

INT247-PROJECT-REPORT

MACHINE LEARNING FOUNDATION

Submitted By:

Name: - Prabhat Kumar

Registration no.: - 11911968

Submitted To:

Faculty: - Dr. Sagar Pande

Date; - 24th/03/2022

List of Contents

ACKNOWLEDGMENT

Special thanks to our Machine learning foundation (INT-247) teacher, Sagar Pande who gave support to us and internet sources which provides all useful references related to this project and who gave us the golden opportunity to do this project and help us in completing the project.

ABSTRACT

Calories in the foods we eat provide energy in the form of heat so that our bodies can function. This means that we need to eat a certain amount of calories just to sustain life. But if we take in too many calories, then we risk gaining weight. So, there is need to burn Calories, for burning calories we doing exercises and more. for know how much calories we have burn Today we are going to buid a machine learning model that predict calories based on some data.

INTRODUCTION

In this fast and busy schedule life, people are not giving importance to the quality of food they are eating. They tend to neglect their eating patterns and habits. The fast-food consumption rate is alarmingly high and this consequently has led to the intake of unhealthy food. This leads to various health issues such as obesity, diabetes, an increase in blood pressure etc. Hence it has become very essential for people to have a good balanced nutritional healthy diet. There are many applications which are booming to help people so that they can have control over their diet and hence can reduce weight or they can help them to keep them fit and healthy. This project focuses on the calories burned in accordance with the duration provided and heart rate during the exercise period. It introduces the topic of linear regression and its predicting capability with the effectiveness from the data provided. This research helps in providing the benefits of a machine learning algorithm over predicting the calories burned.

BENEFITS OF REGULAR PHYSICAL ACTIVITY

1. Exercise controls weights

   Exercise can help prevent excess weight gain or help maintain weight loss. When you engage in physical activity, you burn calories. The more intense the activity, the more calories you burn.

   Regular trips to the gym are great, but don't worry if you can't find a large chunk of time to exercise every day. Any amount of activity is better than none at all. To reap the benefits of exercise, just get more active throughout your day — take the stairs instead of the elevator or rev up your household chores. Consistency is key.

2. Exercise combats health condition and diseases

   Worried about heart disease? Hoping to prevent high blood pressure? No matter what your current weight is, being active boosts high-density lipoprotein (HDL) cholesterol, the "good" cholesterol, and it decreases unhealthy triglycerides. This one-two punch keeps your blood flowing smoothly, which decreases your risk of cardiovascular diseases.

3. Exercise boots energy

   Winded by grocery shopping or household chores? Regular physical activity can improve your muscle strength and boost your endurance.

   Exercise delivers oxygen and nutrients to your tissues and helps your cardiovascular system work more efficiently. And when your heart and lung health improve, you have more energy to tackle daily chores.

4. Exercise promotes better sleep

Struggling to snooze? Regular physical activity can help you fall asleep faster, get better sleep and deepen your sleep. Just don't exercise too close to bedtime, or you may be too energized to go to sleep.

5. Exercise can be fun

Exercise and physical activity can be enjoyable. They give you a chance to unwind, enjoy the outdoors or simply engage in activities that make you happy. Physical activity can also help you connect with family or friends in a fun social setting.

So take a dance class, hit the hiking trails or join a soccer team. Find a physical activity you enjoy, and just do it. Bored? Try something new, or do something with friends or family.

The Bottom line on exercise

Exercise and physical activity are great ways to feel better, boost your health and have fun. For most healthy adults, the U.S. Department of Health and Human Services recommends these exercise guidelines.

- Aerobic Activity: Get at least 150 minutes of moderate aerobic activity or 75 minutes of vigorous aerobic activity a week, or a combination of moderate and vigorous activity. The guidelines suggest that you spread out this exercise during the course of a week. To provide even greater health benefit and to assist with weight loss or maintaining weight loss, at least 300 minutes a week is recommended. But even small amounts of physical activity are helpful. Being active for short periods of time throughout the day can add up to provide health benefit.

- Strength training: Do strength training exercises for all major muscle groups at least two times a week. Aim to do a single set of each exercise using a weight or resistance level heavy enough to tire your muscles after about 12 to 15 repetitions.

  Moderate aerobic exercise includes activities such as brisk walking, biking, swimming and mowing the lawn. Vigorous aerobic exercise includes activities such as running, heavy yardwork and aerobic dancing. Strength training can include use of weight machines, your own body weight, heavy bags, resistance tubing or resistance paddles in the water, or activities such as rock climbing.

BENEFITS OF REGULAR PHYSICAL ACTIVITY

1. Exercise controls weights

Exercise can help prevent excess weight gain or help maintain weight loss. When you engage in physical activity, you burn calories. The more intense the activity, the more calories you burn.

Regular trips to the gym are great, but don't worry if you can't find a large chunk of time to exercise every day. Any amount of activity is better than none at all. To reap the benefits of exercise, just get more active throughout your day — take the stairs instead of the elevator or rev up your household chores. Consistency is key.

2. Exercise combats health condition and diseases

Worried about heart disease? Hoping to prevent high blood pressure? No matter what your current weight is, being active boosts high-density lipoprotein (HDL) cholesterol, the "good" cholesterol, and it decreases unhealthy triglycerides. This one-two punch keeps your blood flowing smoothly, which decreases your risk of cardiovascular diseases.

3. Exercise boots energy

Winded by grocery shopping or household chores? Regular physical activity can improve your muscle strength and boost your endurance.

Exercise delivers oxygen and nutrients to your tissues and helps your cardiovascular system work more efficiently. And when your heart and lung health improve, you have more energy to tackle daily chores.

4. Exercise promotes better sleep

Struggling to snooze? Regular physical activity can help you fall asleep faster, get better sleep and deepen your sleep. Just don't exercise too close to bedtime, or you may be too energized to go to sleep.

5. Exercise can be fun

Exercise and physical activity can be enjoyable. They give you a chance to unwind, enjoy the outdoors or simply engage in activities that make you happy. Physical activity can also help you connect with family or friends in a fun social setting.

So take a dance class, hit the hiking trails or join a soccer team. Find a physical activity you enjoy, and just do it. Bored? Try something new, or do something with friends or family.

The Bottom line on exercise

Exercise and physical activity are great ways to feel better, boost your health and have fun. For most healthy adults, the U.S. Department of Health and Human Services recommends these exercise guidelines.

- Aerobic Activity: Get at least 150 minutes of moderate aerobic activity or 75 minutes of vigorous aerobic activity a week, or a combination of moderate and vigorous activity. The guidelines suggest that you spread out this exercise during the course of a week. To provide even greater health benefit and to assist with weight loss or maintaining weight loss, at least 300 minutes a week is recommended. But even small amounts of physical activity are helpful. Being active for short periods of time throughout the day can add up to provide health benefit.

- Strength training: Do strength training exercises for all major muscle groups at least two times a week. Aim to do a single set of each exercise using a weight or resistance level heavy enough to tire your muscles after about 12 to 15 repetitions.

Moderate aerobic exercise includes activities such as brisk walking, biking, swimming and mowing the lawn. Vigorous aerobic exercise includes activities such as running, heavy yardwork and aerobic dancing. Strength training can include use of weight machines, your own body weight, heavy bags, resistance tubing or resistance paddles in the water, or activities such as rock climbing

Data description

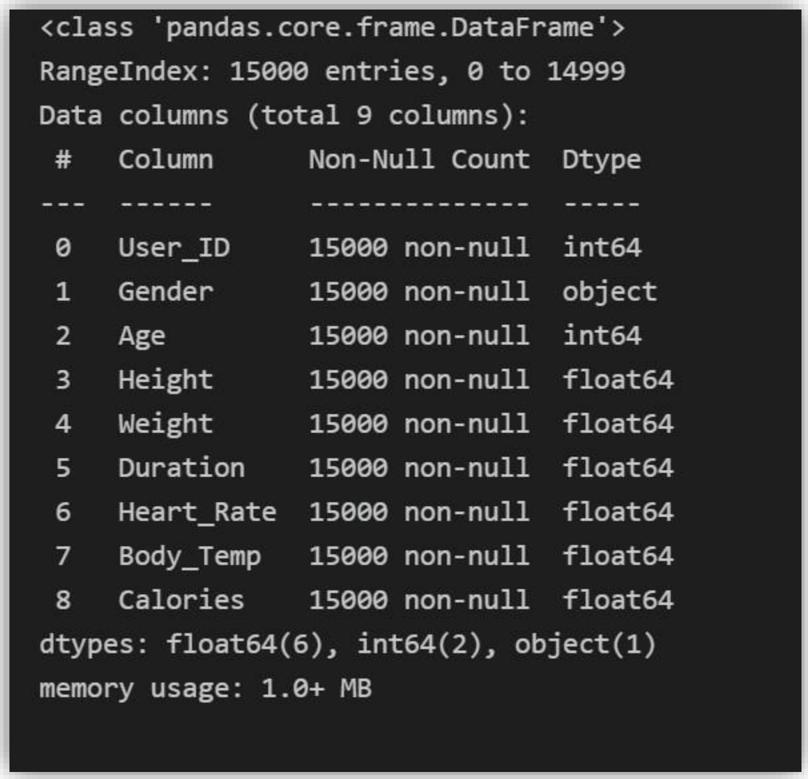
In this I have used two data files one is exercise.csv and calories.csv

Exercise.csv file contain data about person it contain person’s

  - User_ID
  - Gender
  - Age
  - Height
  - Weight
  - Duration (exercise)
  - Heart_ rate Average
  - Body_Temperature

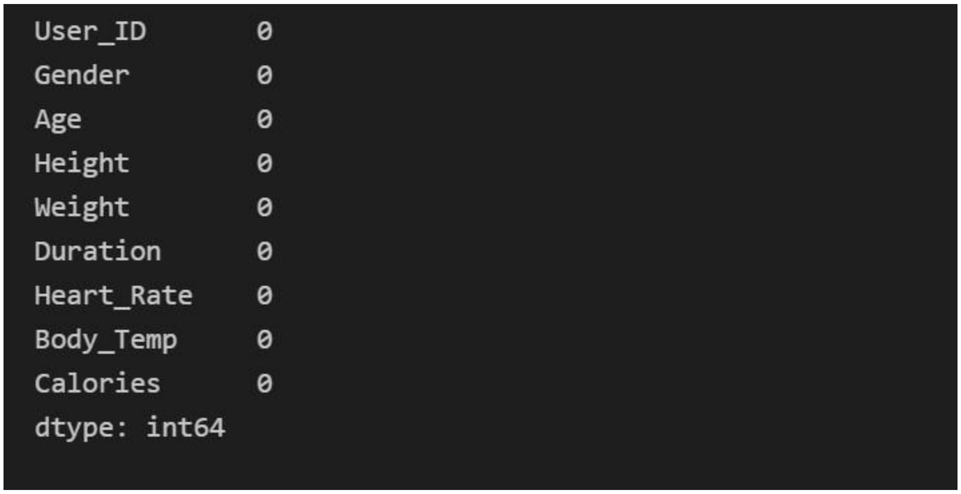
Calories.csv file contain :-

- User_ID
- Calories



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   User_ID     15000 non-null  int64
 1   Gender      15000 non-null  object
 2   Age         15000 non-null  int64
 3   Height      15000 non-null  float64
 4   Weight      15000 non-null  float64
 5   Duration    15000 non-null  float64
 6   Heart_Rate  15000 non-null  float64
 7   Body_Temp   15000 non-null  float64
 8   Calories    15000 non-null  float64
dtypes: float64(6), int64(2), object(1)
memory usage: 1.0+ MB
```

Data about null values:-



```
User_ID       0
Gender        0
Age           0
Height        0
Weight        0
Duration      0
Heart_Rate    0
Body_Temp     0
Calories      0
dtype: int64
```

TO HANDLE MISSING VALUES IN DATASET

In real world data, there are some instances where a particular element is absent because of various reasons, such as, corrupt data, failure to load the information, or incomplete extraction. Handling the missing values is one of the greatest challenges faced by analysts, because making the right decision on how to handle it generates robust data models. Let us look at different ways of imputing the missing values

1. Deleting Rows

   This method commonly used to handle the null values. Here, we either delete a particular row if it has a null value for a particular feature and a particular column if it has more than 70-75% of missing values. This method is advised only when there are enough samples in the data set. One has to make sure that after we have deleted the data, there is no addition of bias. Removing the data will lead to loss of information which will not give the expected results while predicting the output.
2. Replacing With Mean/Median/Mode

   This strategy can be applied on a feature which has numeric data like the age of a person or the ticket fare. We can calculate the mean, median or mode of the feature and replace it with the missing values. This is an approximation which can add variance to the data set. But the loss of the data can be negated by this method which yields better results compared to removal of rows and columns. Replacing with the above three approximations are a statistical approach of handling the missing values. This method is also called as leaking the data while training. Another way is to approximate it with the deviation of neighbouring values. This works better if the data is linear.

3. Assigning an unique category

   A categorical feature will have a definite number of possibilities, such as gender, for example. Since they have a definite number of classes, we can assign another class for the missing values. Here, the features Cabin and Embarked have missing values which can be replaced with a new category, say, U for ‘unknown’. This strategy will add more information into the dataset which will result in the change of variance. Since they are categorical, we need to find one hot encoding to convert it to a numeric form for the algorithm to understand it.

4. Predicting the Missing values

   Using the features which do not have missing values, we can predict the nulls with the help of a machine learning algorithm. This method may result in better accuracy, unless a missing value is expected to have a very high variance. We will be using linear regression to replace the nulls in the feature ‘age’, using other available features. One can experiment with different algorithms and check which gives the best accuracy instead of sticking to a single algorithm.

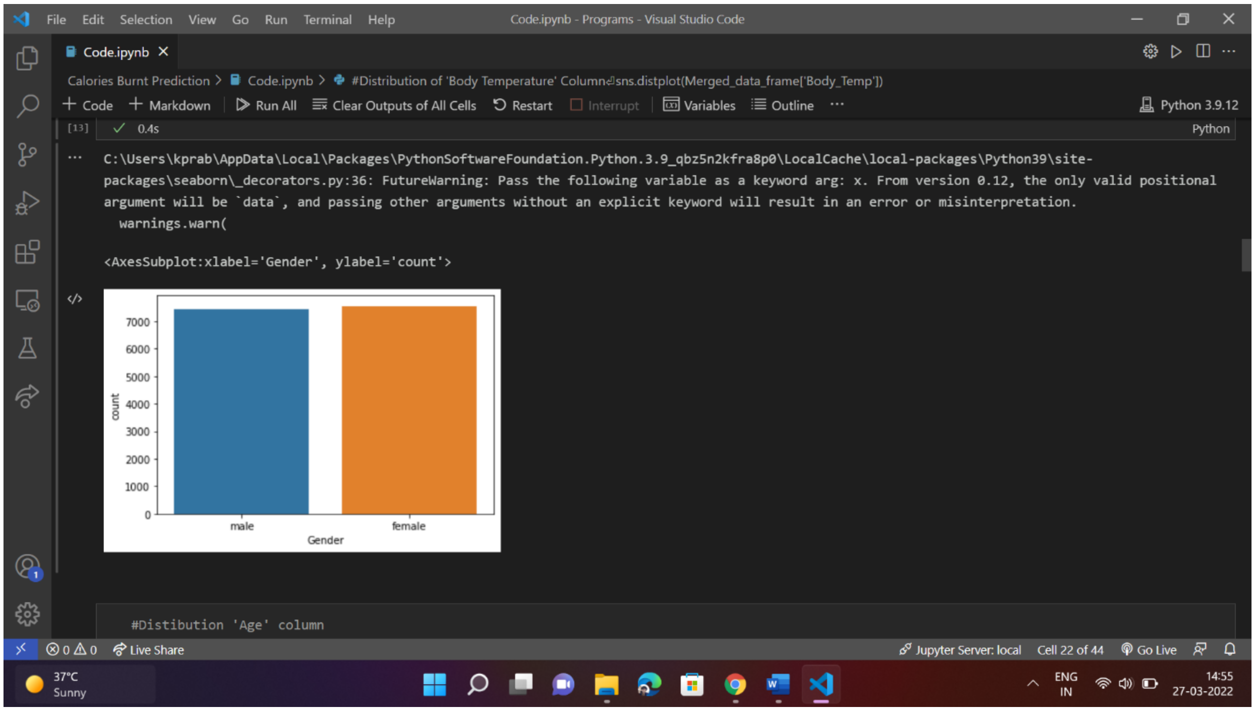
5. Using Algorithms which support missing values

   KNN is a machine learning algorithm which works on the principle of distance measure. This algorithm can be used when there are nulls present in the dataset. While the algorithm is applied, KNN considers the missing values by taking the majority of the K nearest values. In this particular dataset, taking into account the person's age, sex, class etc, we will assume that people having same data for the above mentioned features will have the same kind of fare.

   But is good we don't have to use any one of it because we have don't have any missing or null value in dataset.

Data Visualization of some features

1) Gender Column:

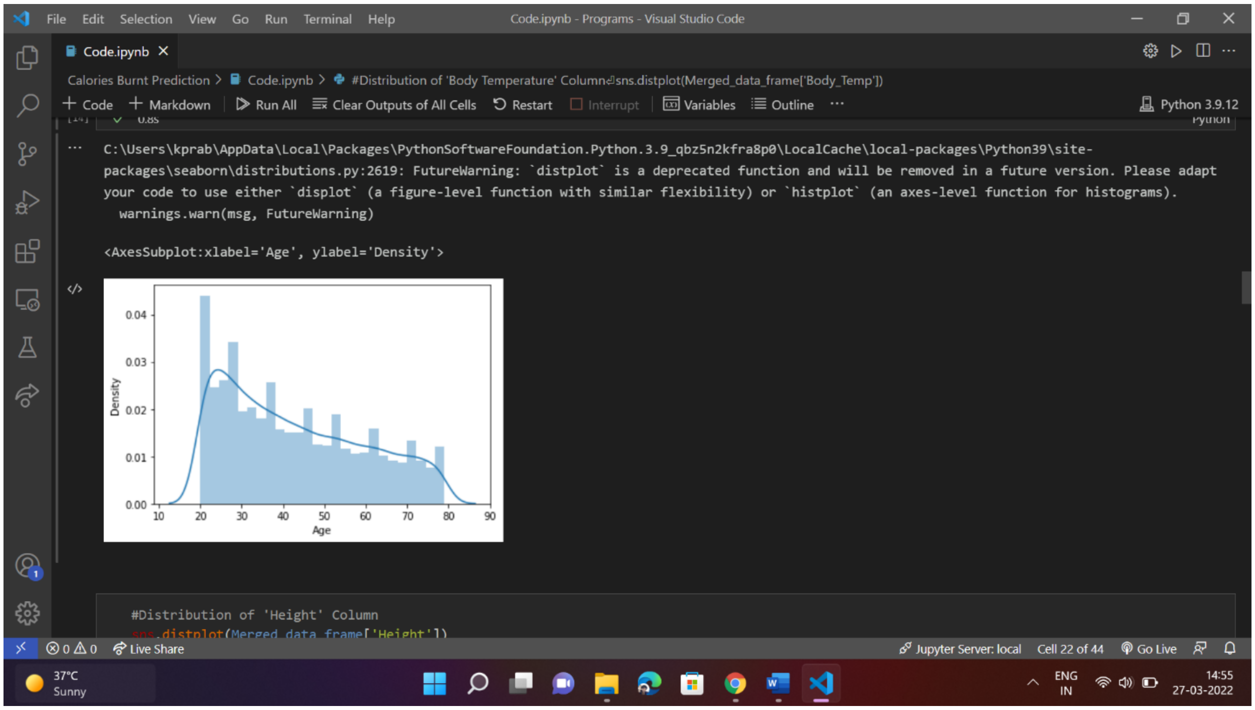


```
C:\Users\kprab\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-
packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional
argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(

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

```
#Distibution 'Age' column
```

2) Age column

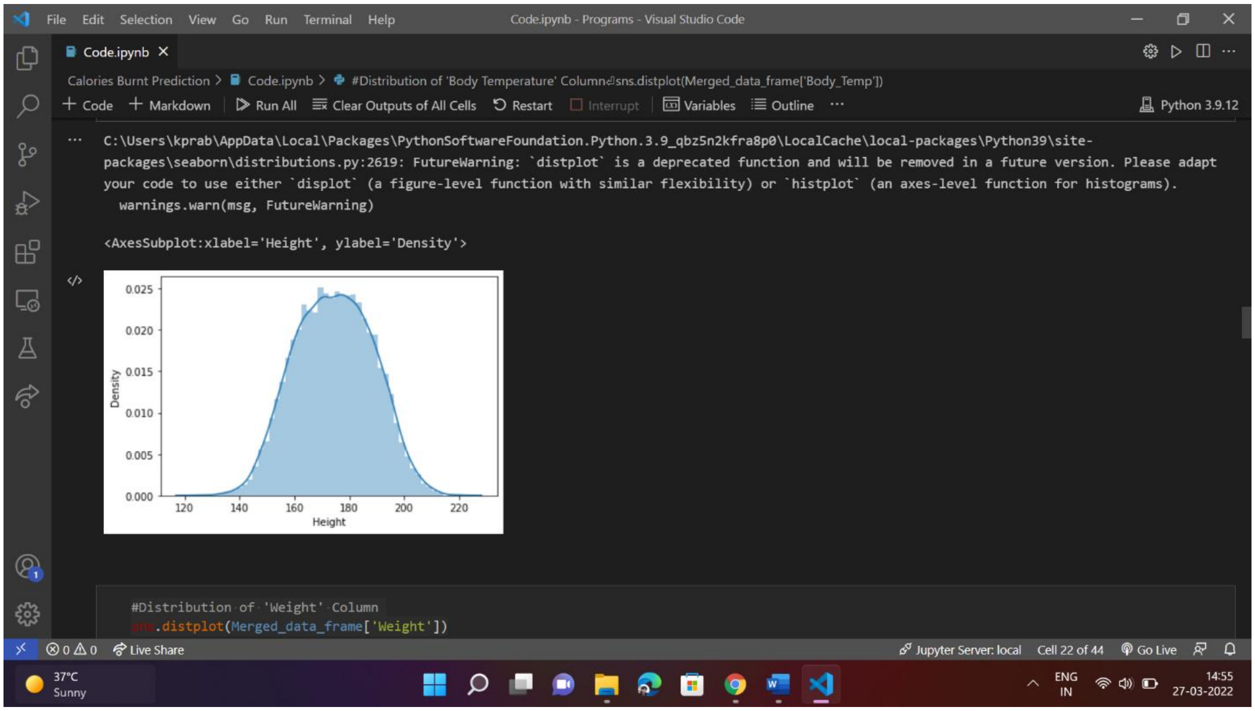


```
C:\Users\kprab\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt
your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Age', ylabel='Density'>
```

```
#Distribution of 'Height' Column
sns.distplot(Merged_data_frame['Height'])
```

3) Height Column

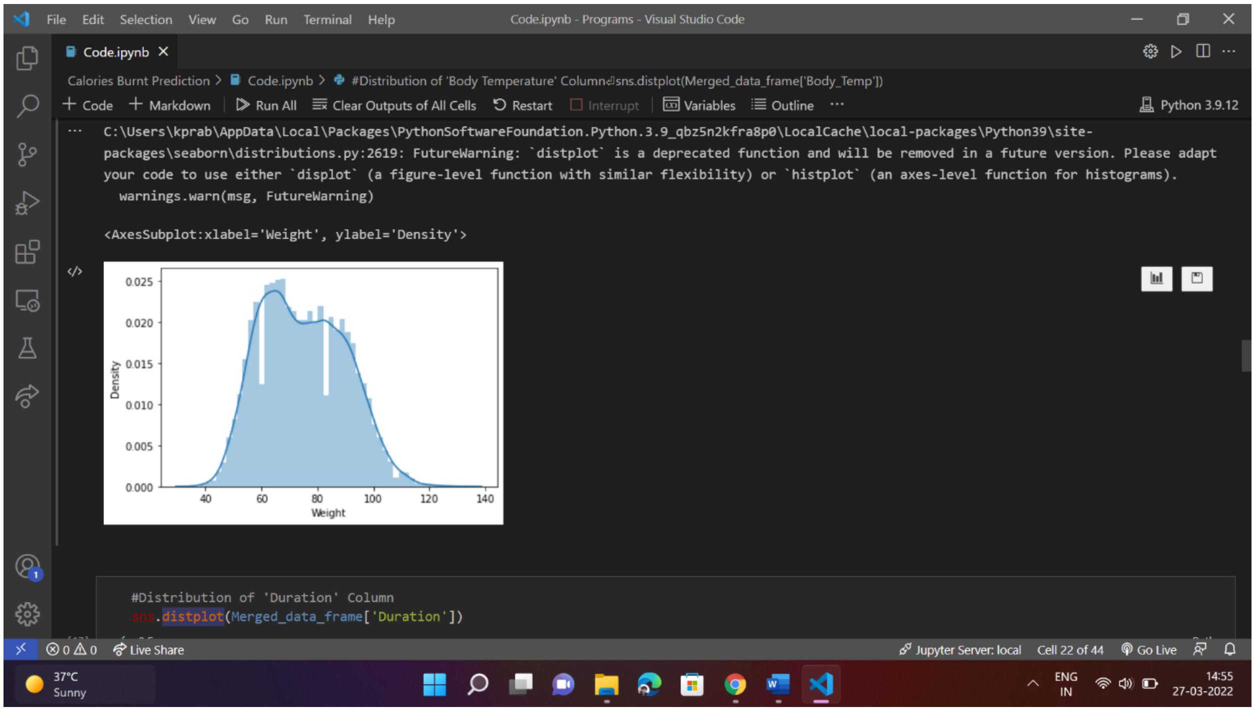

File Edit Selection View Go Run Terminal Help
Code.ipynb - Programs - Visual Studio Code
Code.ipynb
Calories Burnt Prediction > Code.ipynb > #Distribution of 'Body Temperature' Column⏎sns.distplot(Merged_data_frame['Body_Temp'])
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.9.12
0.025
0.020
0.015
0.010
0.005
0.000
Density
120
140
160
180
200
220
Height
Jupyter Server: local Cell 22 of 44 Go Live
Live Share
37°C
Sunny
ENG
IN
14:55
27-03-2022

```
C:\Users\kprab\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt
your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Height', ylabel='Density'>
```

```
#Distribution of 'Weight' Column
sns.distplot(Merged_data_frame['Weight'])
```

4) Weight Column

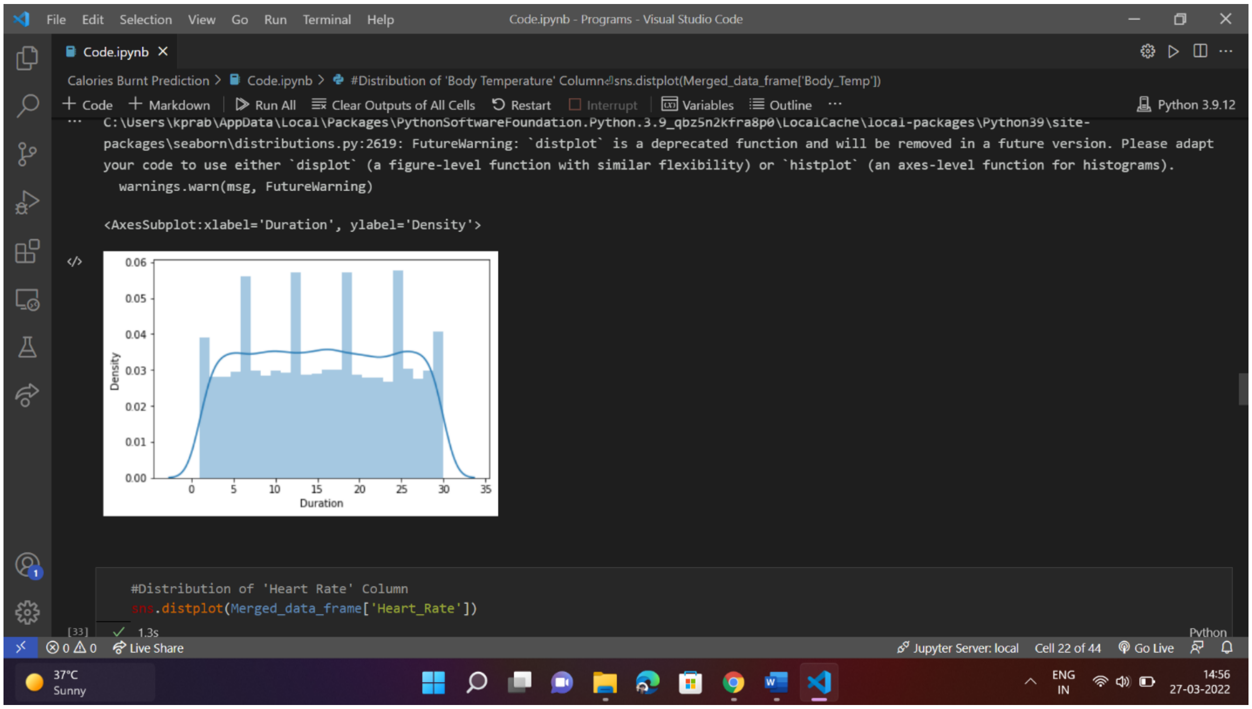

File Edit Selection View Go Run Terminal Help
Code.ipynb - Programs - Visual Studio Code
Code.ipynb
Calories Burnt Prediction > Code.ipynb > #Distribution of 'Body Temperature' Column⏎sns.distplot(Merged_data_frame['Body_Temp'])
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.9.12
0.025
0.020
0.015
0.010
0.005
0.000
Density
40
60
80
100
120
140
Weight
Jupyter Server: local Cell 22 of 44 Go Live
Live Share
37°C
Sunny
ENG
IN
14:55
27-03-2022

```
C:\Users\kprab\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt
your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Weight', ylabel='Density'>
```

```
#Distribution of 'Duration' Column
sns.distplot(Merged_data_frame['Duration'])
```

5) Duration Column

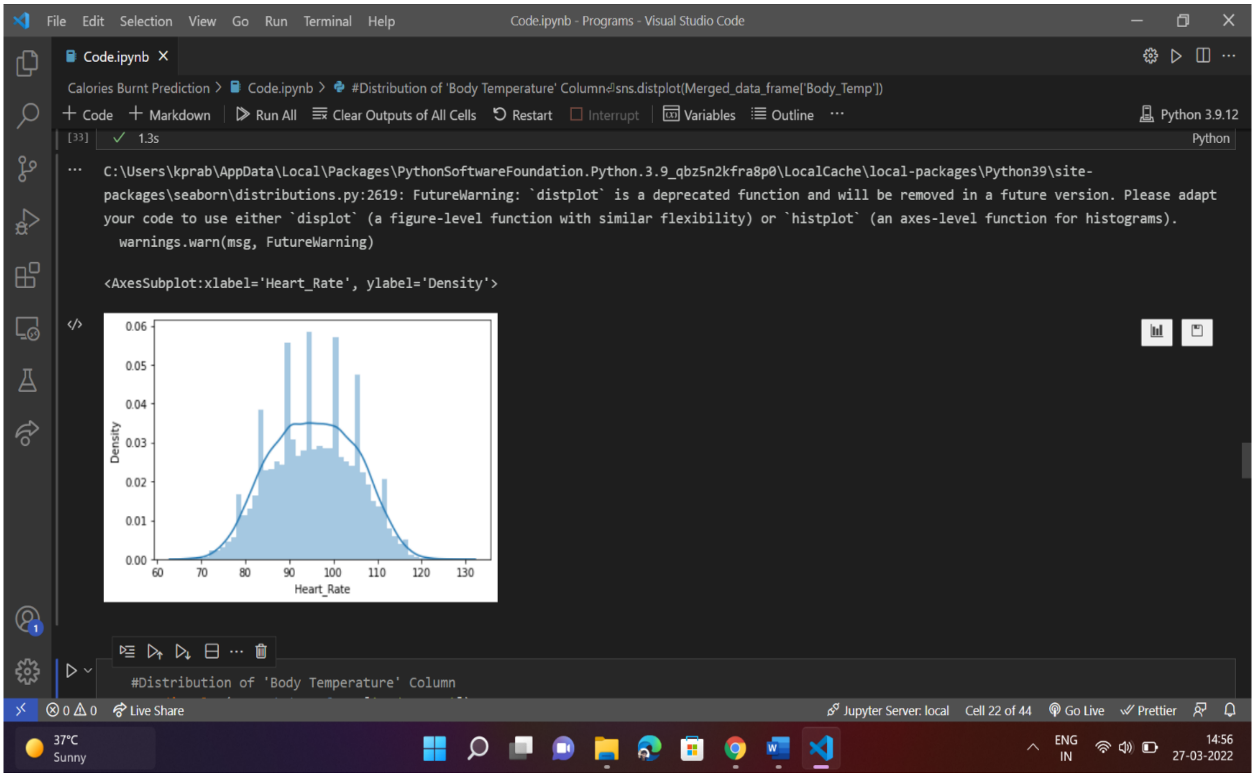

File Edit Selection View Go Run Terminal Help
Code.ipynb - Programs - Visual Studio Code
Code.ipynb
Calories Burnt Prediction > Code.ipynb > #Distribution of 'Body Temperature' Column⏎sns.distplot(Merged_data_frame['Body_Temp'])
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.9.12
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
0
5
10
15
20
25
30
35
Duration
[33]
1.3s
Python
0 0
Live Share
Jupyter Server: local
Cell 22 of 44
Go Live
37°C
Sunny
ENG
IN
14:56
27-03-2022

```
C:\Users\kprab\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt
your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Duration', ylabel='Density'>
```

```
#Distribution of 'Heart Rate' Column
sns.distplot(Merged_data_frame['Heart_Rate'])
```

6) Heart Rate

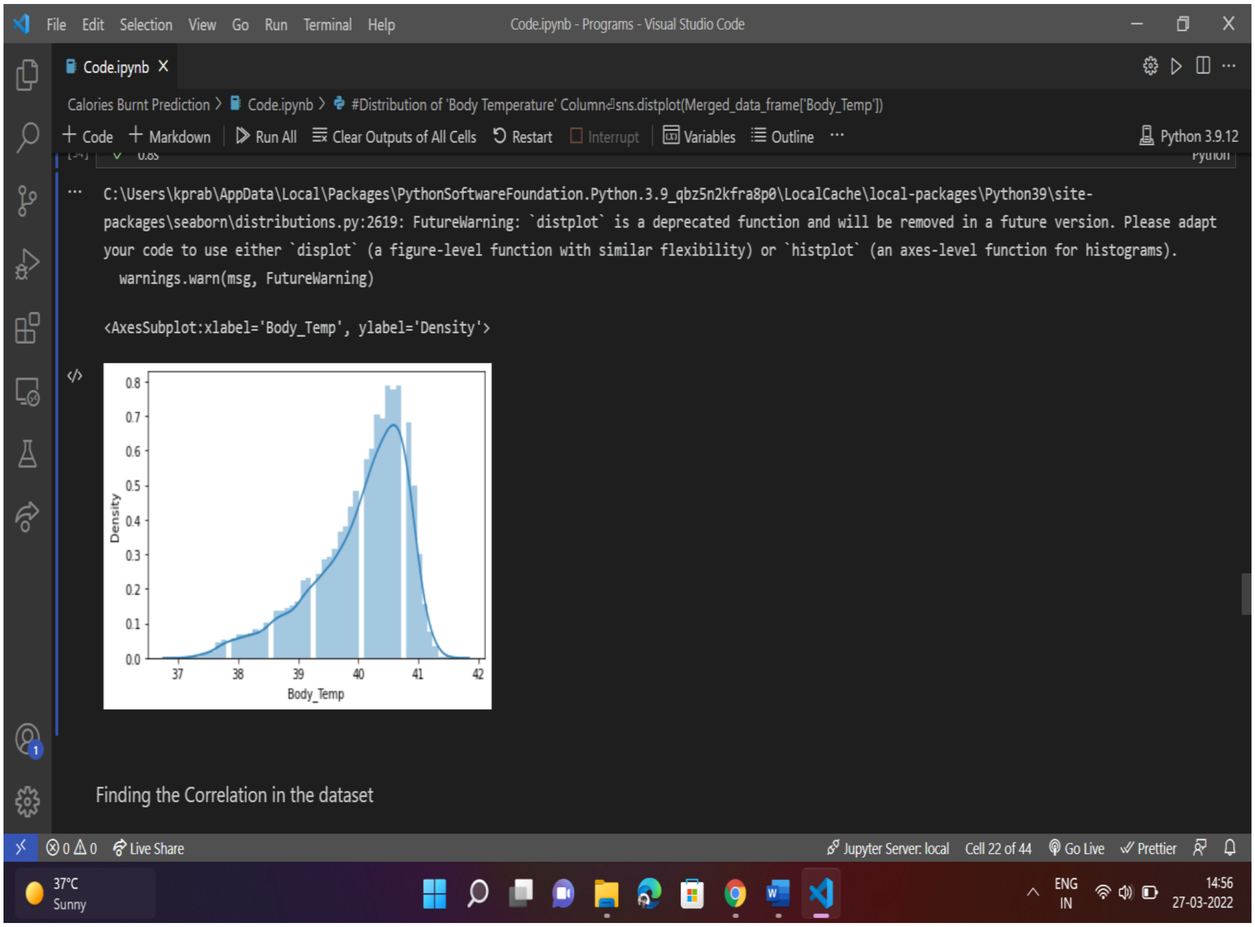

File Edit Selection View Go Run Terminal Help
Code.ipynb - Programs - Visual Studio Code
Code.ipynb
Calories Burnt Prediction > Code.ipynb > #Distribution of 'Body Temperature' Column⏎sns.distplot(Merged_data_frame['Body_Temp'])
+ Code + Markdown Run All Clear Outputs of All Cells Restart Interrupt Variables Outline
Python 3.9.12
[33]
1.3s
Python
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
60
70
80
90
100
110
120
130
Heart_Rate
0 0
Live Share
Jupyter Server: local
Cell 22 of 44
Go Live
Prettier
37°C
Sunny
ENG
IN
14:56
27-03-2022

```
C:\Users\kprab\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt
your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Heart_Rate', ylabel='Density'>
```

```
#Distribution of 'Body Temperature' Column
```

7) Body Temperature



```
C:\Users\kprab\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.9_qbz5n2kfra8p0\LocalCache\local-packages\Python39\site-
packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt
your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='Body_Temp', ylabel='Density'>
```

Finding the Correlation in the dataset

CORRELATION IN DATASET

Machine Learning models are as good or as bad as the data you have. That’s why data scientists can spend hours on pre-processing and cleansing the data. They select only the features that would contribute most to the quality of the resulting model. This process is called “Feature Selection”. Feature Selection is the process of selecting the attributes that can make the predicted variable more accurate or eliminating those attributes that are irrelevant and can decrease the model accuracy and quality.

Data and feature correlation is considered one important step in the feature selection phase of the data pre-processing especially if the data type for the features is continuous.

Data Correlation: Is a way to understand the relationship between multiple variables and attributes in your dataset. Using Correlation, you can get some insights such as:

- One or multiple attributes depend on another attribute or a cause for another attribute.
- One or multiple attributes are associated with other attributes.

Why Correlation is useful?

- Correlation can help in predicting one attribute from another (Great way to impute missing values).
- Correlation can (sometimes) indicate the presence of a causal relationship.
- Correlation is used as a basic quantity for many modelling techniques.

The performance of some algorithms can deteriorate if two or more variables are tightly related, called multicollinearity. An example is linear regression, where one of the offending correlated variables should be removed in order to improve the skill of the model.

We may also be interested in the correlation between input variables with the output variable in order provide insight into which variables may or may not be relevant as input for developing a model.The structure of the relationship may be known, e.g. it may be linear, or we may have no idea whether a relationship exists between two variables or what structure it may take. Depending what is known about the relationship and the distribution of the variables, different correlation scores can be calculated.

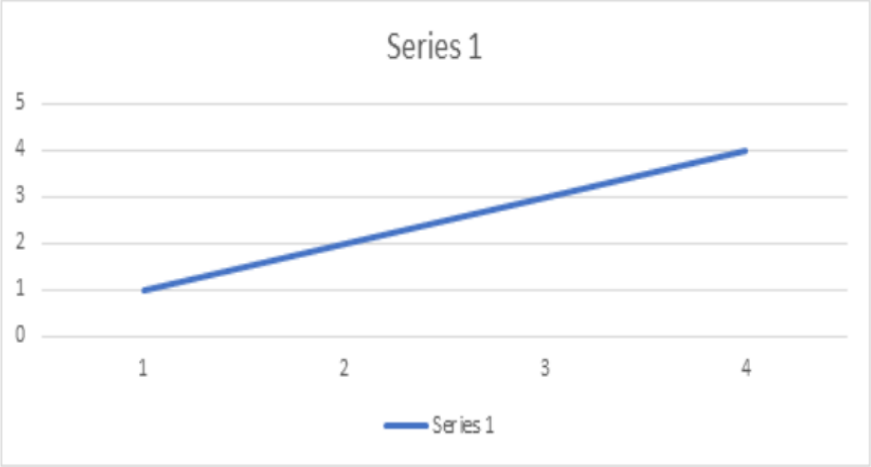
Types Of Correlation:-

1) Positive Correlation:-

Two features (variables) can be positively correlated with each other. It means that when the value of one variable increase then the value of the other variable(s) also increases

A positive correlation does not guarantee growth or benefit. Instead, it is used to denote any two or more variables that move in the same direction together, so when one increases, so does the other. But the existence of a correlation does not necessarily indicate a causal relationship between variables.

Correlation is a form of dependency, where a shift in one variable means a change is likely in the other, or that certain known variables produce specific results. A general example can be seen within complementary product demand. If the demand for vehicles rises, so will the demand for vehicular-related products and services, such as tires. An increase in one area has an effect on complementary industries.

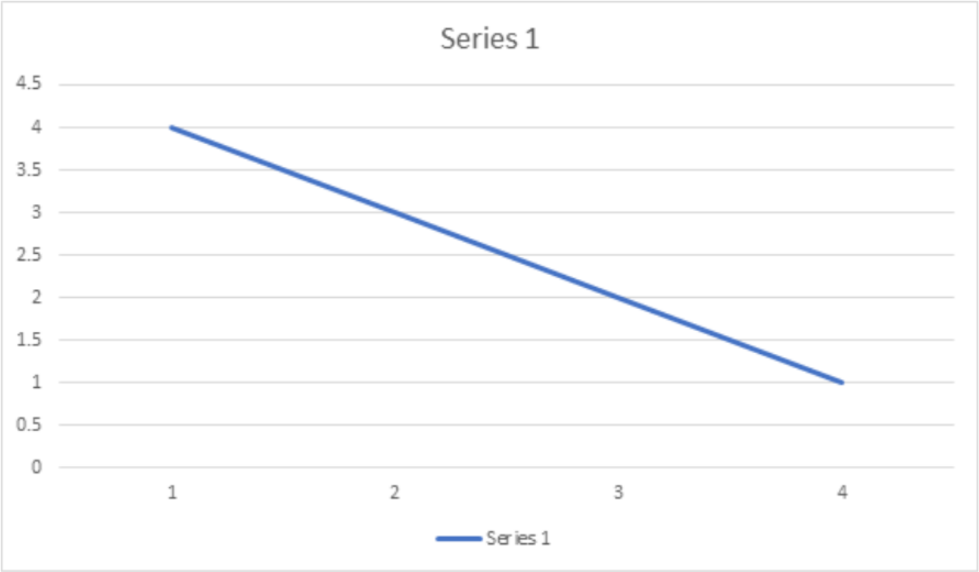


Positive Correlation

2) Negative Correlation:

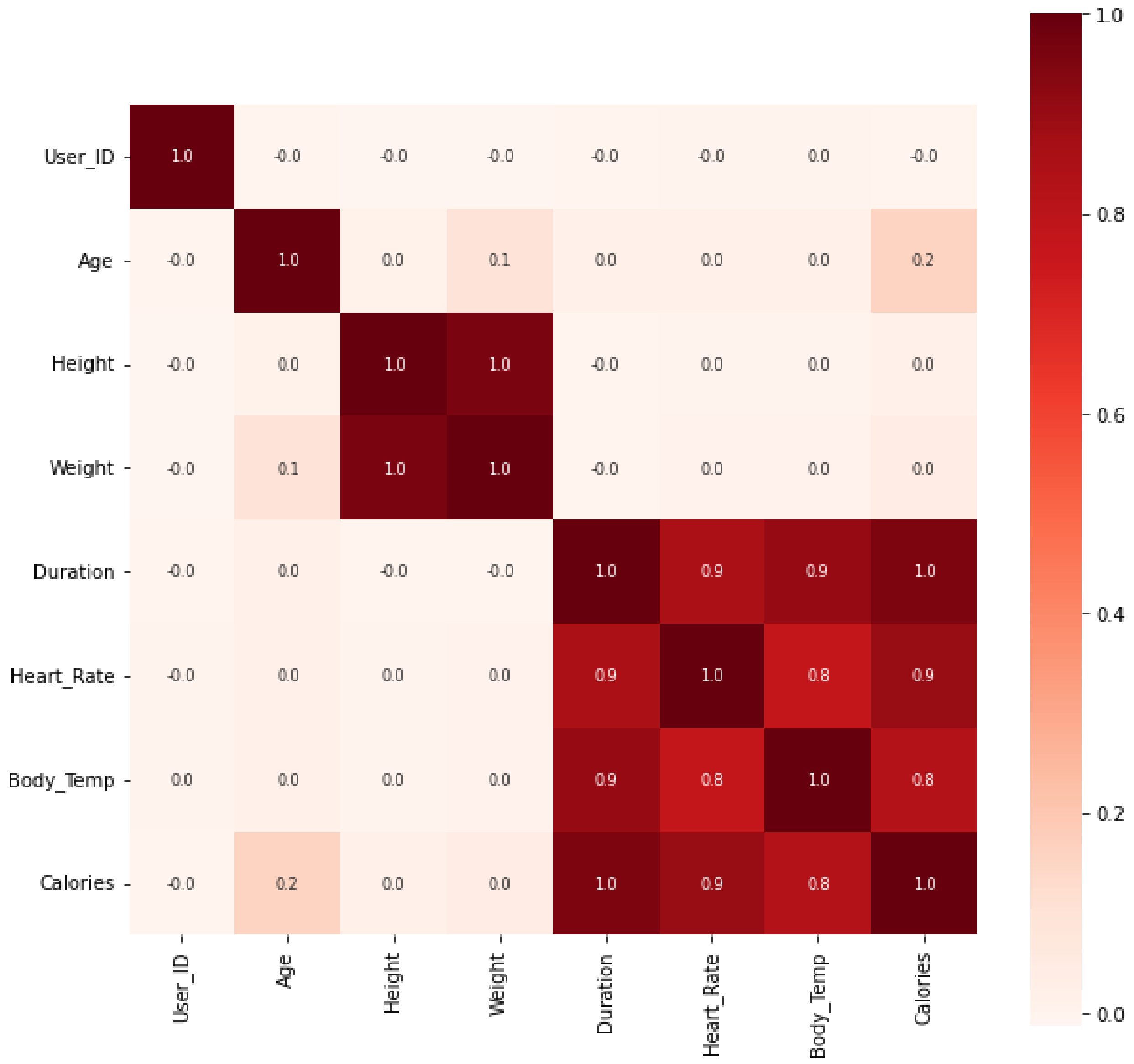
Two features (variables) can be negatively correlated with each other. It means that when the value of one variable increase then the value of the other variable(s) decreases.

In terms of machine learning a strong correlation (either positive or negative) is good news. Your classifier has a relation between two variables that it can work with. No correlation (zero, 0) is bad, since there doesn't seem a relation between the features.



Negative Correlation

- By using heat map I show the correlation between features of our dataframe:


User_ID
Age
Height
Weight
Duration
Heart_Rate
Body_Temp
Calories
1.0 -0.0 -0.0 -0.0 -0.0 -0.0 0.0 -0.0
-0.0 1.0 0.0 0.1 0.0 0.0 0.0 0.2
-0.0 0.0 1.0 1.0 -0.0 0.0 0.0 0.0
-0.0 0.1 1.0 1.0 -0.0 0.0 0.0 0.0
-0.0 0.0 -0.0 -0.0 1.0 0.9 0.9 1.0
-0.0 0.0 0.0 0.0 0.9 1.0 0.8 0.9
0.0 0.0 0.0 0.0 0.9 0.8 1.0 0.8
-0.0 0.2 0.0 0.0 1.0 0.9 0.8 1.0
1.0
0.8
0.6
0.4
0.2
0.0

Model Training

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.

This iterative process is called “model fitting”. The accuracy of the training dataset or the validation dataset is critical for the precision of the model. Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved. There are several types of machine learning models, of which the most common ones are supervised and unsupervised learning. Supervised learning is possible when the training data contains both the input and output values. Each set of data that has the inputs and the expected output is called a supervisory signal. The training is done based on the deviation of the processed result from the documented result when the inputs are fed into the model. Unsupervised learning involves determining patterns in the data. Additional data is then used to fit patterns or clusters. This is also an iterative process that improves the accuracy based on the correlation to the expected patterns or clusters. There is no reference output dataset in this method.

Linear regression finds the linear relationship between the dependent variable and one or more independent variables using a best-fit straight line.

Generally, a linear model makes a prediction by simply computing a weighted sum of the input features, plus a constant called the bias term (also called the intercept term). In this technique, the dependent variable is continuous, the independent variable(s) can be continuous or discrete, and the nature of the regression line is linear.

Gradient descent is an optimization technique used to tune the coefficient and bias of a linear equation.

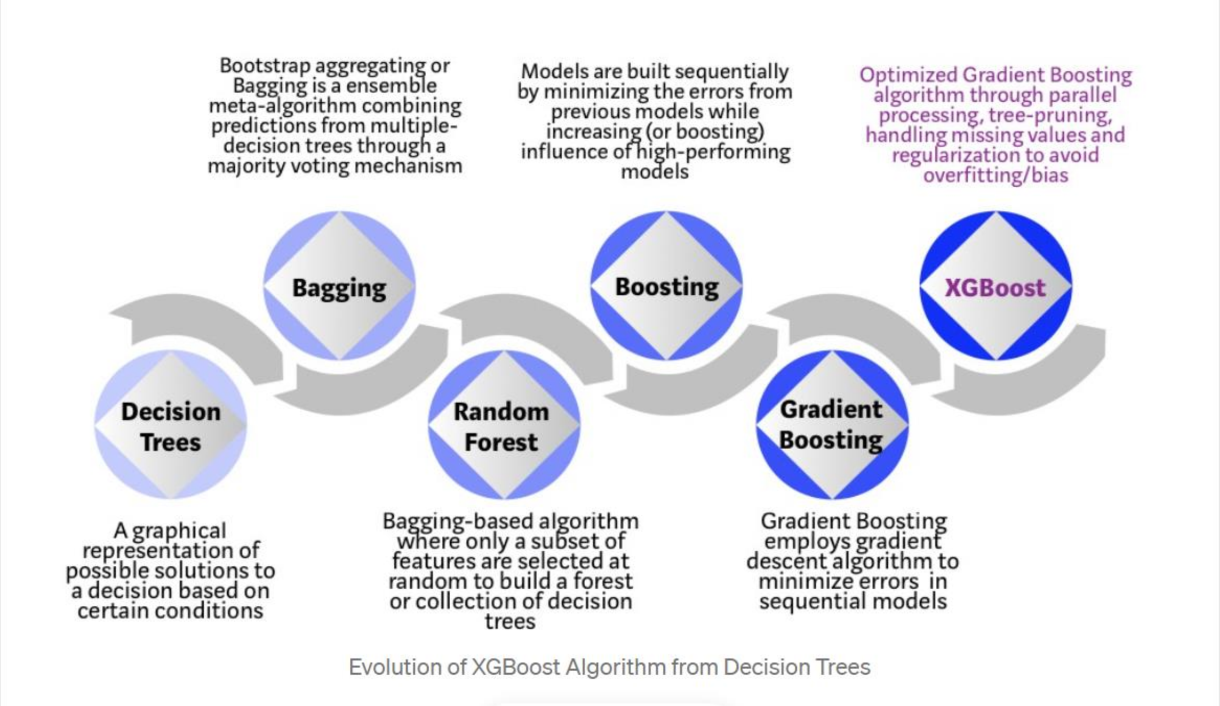
Imagine you are on the top left of a u-shaped cliff and moving blind-folded towards the bottom center. You take small steps in the direction of the steepest slope. This is what gradient descent does — it is the derivative or the tangential line to a function that attempts to find local minima of a function.

We take steps down the cost function in the direction of the steepest descent until we reach the minima, which in this case is the downhill. The size of each step is determined by the parameter α, called learning rate. If it's too big, the model might miss the local minimum of the function, and if it's too small, the model will take a long time to converge. Hence, α provides the basis for finding the local minimum, which helps in finding the minimized cost function.

‘Q’ the cost function is differentiated w.r.t the parameters, m and c to arrive at the updated m and c, respectively. The product of the differentiated value and learning rate is subtracted from the actual ones to minimize the parameters affecting the model.

XGBoost Model

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now. Please see the chart below for the evolution of tree-based algorithms over the years.


Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism
Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias
Bagging
Boosting
XGBoost
Decision Trees
Random Forest
Gradient Boosting
A graphical representation of possible solutions to a decision based on certain conditions
Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees
Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models

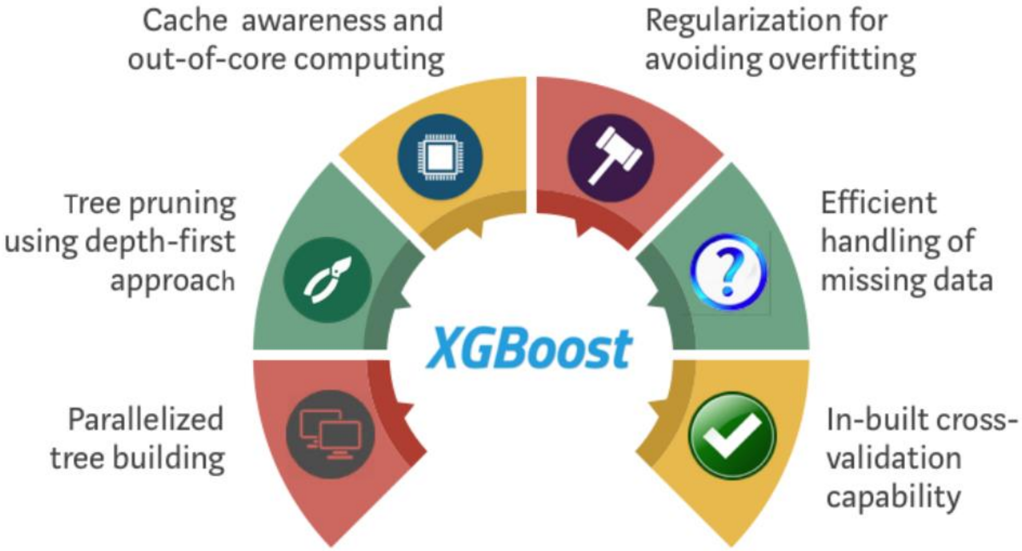
Evolution of XGBoost Algorithm from Decision Trees

XGBoost algorithm was developed as a research project at the University of Washington. Tianqi Chen and Carlos Guestrin presented their paper at SIGKDD Conference in 2016 and caught the Machine Learning world by fire. Since its introduction, this algorithm has not only been credited with winning numerous Kaggle competitions but also for being the driving force under the hood for several cutting-edge industry applications. As a result, there is a strong community of data scientists contributing to the XGBoost open source projects with ~350 contributors and ~3,600 commits on GitHub. The algorithm differentiates itself in the following ways:

1. A wide range of applications: Can be used to solve regression, classification, ranking, and user-defined prediction problems.
2. Portability: Runs smoothly on Windows, Linux, and OS X.
3. Languages: Supports all major programming languages including C++, Python, R, Java, Scala, and Julia.
4. Cloud Integration: Supports AWS, Azure, and Yarn clusters and works well with Flink, Spark, and other ecosystems.

Why Does XGBoost Perform so well?

XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods that apply the principle of boosting weak learners (CARTs generally) using the gradient descent architecture. However, XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements.


Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient handling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

How XGBoost optimizes standard GBM algorithm

Checking Model Accuracy

There are many ways of measuring a model's accuracy. However, the Mean Absolute Error, also known as MAE, is one of the many metrics for summarizing and assessing the quality of a machine learning model.

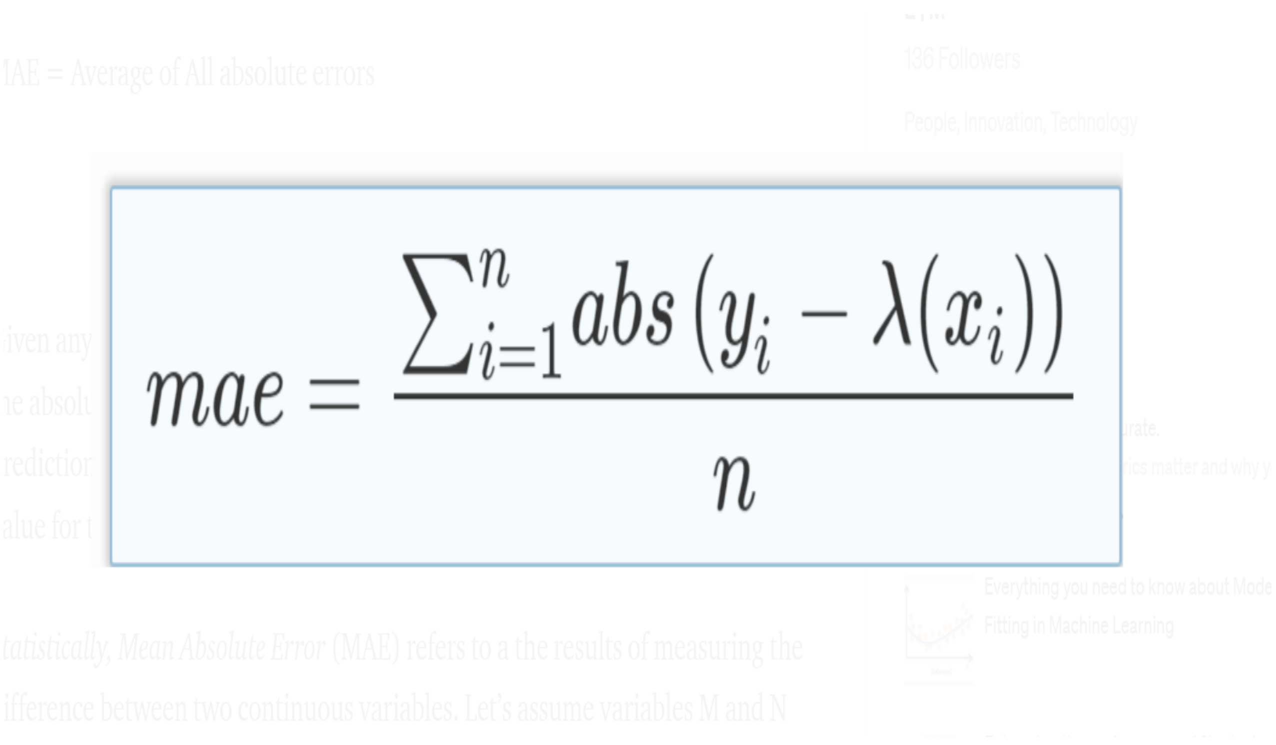
What exactly does 'ERROR' in this metric mean ? We do a subtraction of Predicted value from Actual Value as below.

Prediction Error → Actual Value - Predicted Value

This prediction error is taking for each record after which we convert all error to positive. This is achieved by taking Absolute value for each error as below

Absolute Error → |Prediction Error|

Finally we calculate the mean for all recorded absolute errors (Average sum of all absolute errors).



IAE = Average of All absolute errors

136 Followers

People, Innovation, Technology

$$mae = \frac{\sum_{i=1}^{n} abs\left(y_i - \lambda(x_i)\right)}{n}$$

iven any

ie absolu

rediction

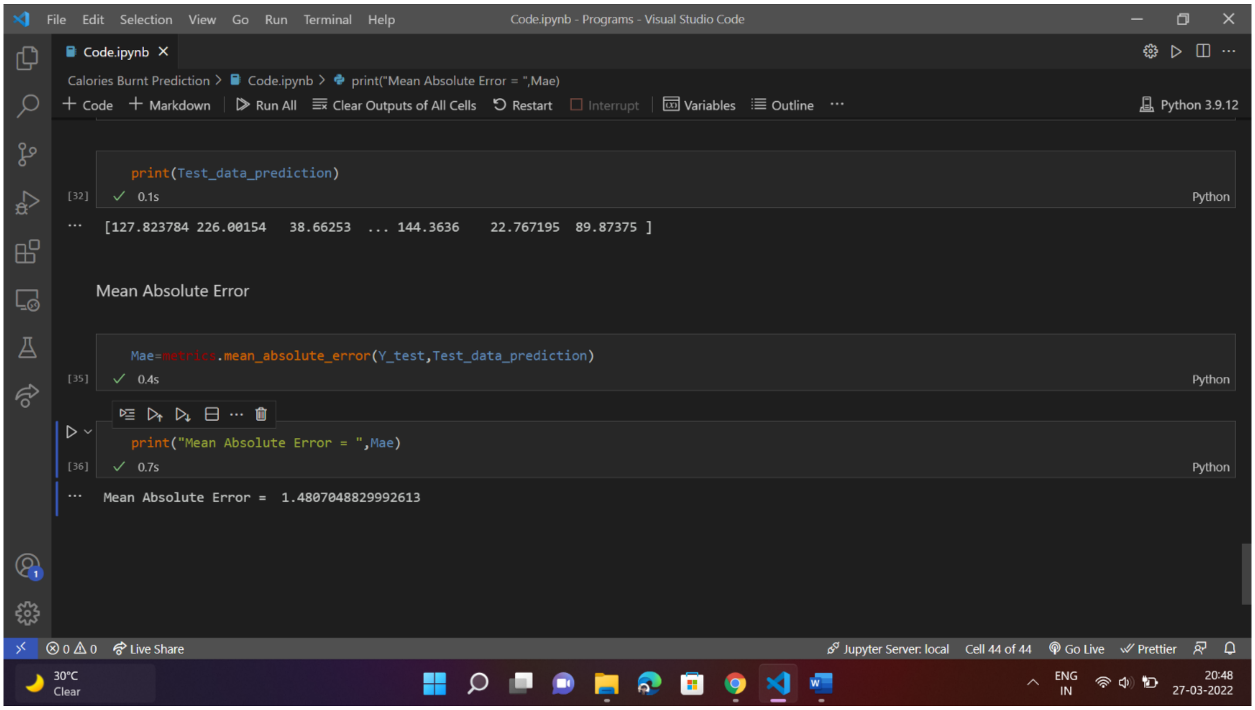
alue for t

urate.

rics matter and why y

Everything you need to know about Mode Fitting in Machine Learning

tatistically, Mean Absolute Error (MAE) refers to a the results of measuring the ifference between two continuous variables. Let's assume variables M and N



```
print(Test_data_prediction)
```

```
[127.823784 226.00154   38.66253  ... 144.3636    22.767195  89.87375 ]
```

Mean Absolute Error

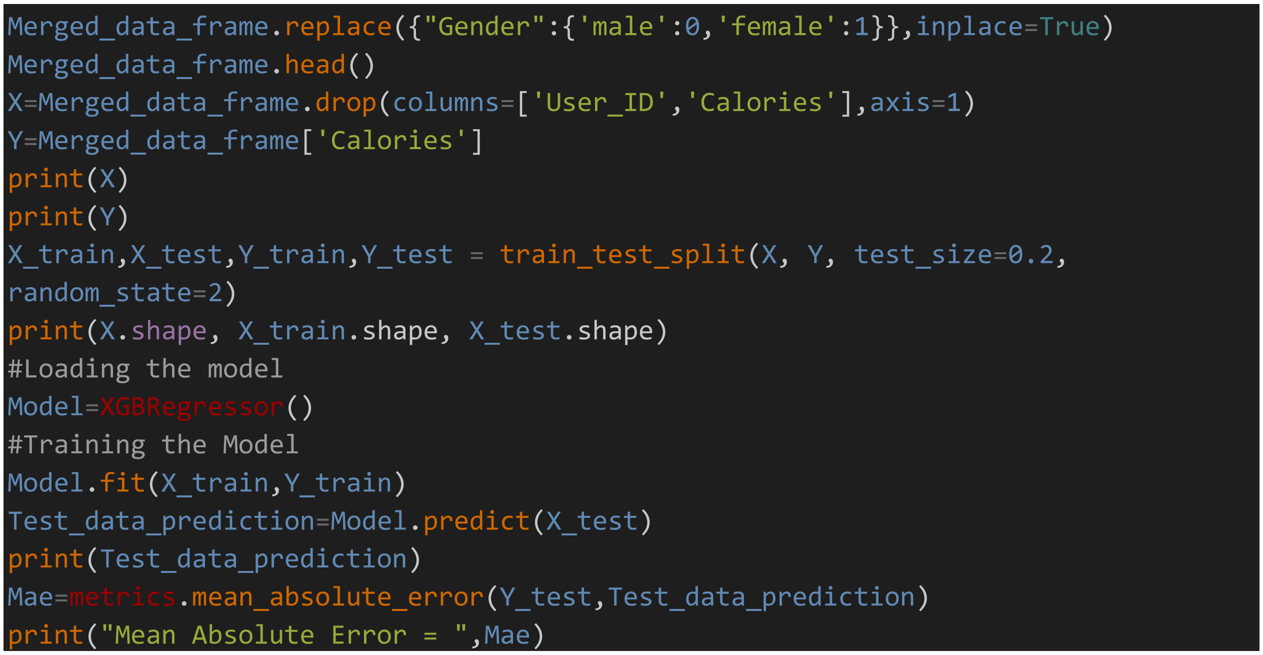
```
Mae=metrics.mean_absolute_error(Y_test,Test_data_prediction)
```

```
print("Mean Absolute Error = ",Mae)
```

```
Mean Absolute Error =  1.4807048829992613
```

Code of Project

```
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
#loading the data from csv file to a Panda Dataframe
calories_data=pd.read_csv('calories.csv')
#print the first 5 rows of the dataframe
calories_data.head()
exercise_data=pd.read_csv('exercise.csv')
exercise_data.head()
Merged_data_frame=pd.concat([exercise_data,calories_data['Calories']],axis=1)
Merged_data_frame.head()
#Checking the number of rows and columns in our Dataframe
Merged_data_frame.shape
#getting some information about the data
Merged_data_frame.info()
#checking for missing values
Merged_data_frame.isnull().sum()
#Get some statistical measures about the data
Merged_data_frame.describe()
#Plotting the gender column in count plot
sns.countplot(Merged_data_frame['Gender'])
#Distibution 'Age' column
sns.distplot(Merged_data_frame['Age'])
#Distribution of 'Height' Column
sns.distplot(Merged_data_frame['Height'])
#Distribution of 'Weight' Column
sns.distplot(Merged_data_frame['Weight'])
#Distribution of 'Duration' Column
sns.distplot(Merged_data_frame['Duration'])
#Distribution of 'Heart Rate' Column
sns.distplot(Merged_data_frame['Heart_Rate'])
#Distribution of 'Body Temperature' Column
sns.distplot(Merged_data_frame['Body_Temp'])
correlation=Merged_data_frame.corr()
#Constructing a heatmap for better understanding of correlation
plt.figure(figsize=(10,10))
sns.heatmap(correlation,cbar=True,square=True,fmt='.1f',annot=True,annot_kws={'si
ze':8},cmap='Reds')
```



```
Merged_data_frame.replace({"Gender":{'male':0,'female':1}},inplace=True)
Merged_data_frame.head()
X=Merged_data_frame.drop(columns=['User_ID','Calories'],axis=1)
Y=Merged_data_frame['Calories']
print(X)
print(Y)
X_train,X_test,Y_train,Y_test = train_test_split(X, Y, test_size=0.2,
random_state=2)
print(X.shape, X_train.shape, X_test.shape)
#Loading the model
Model=XGBRegressor()
#Training the Model
Model.fit(X_train,Y_train)
Test_data_prediction=Model.predict(X_test)
print(Test_data_prediction)
Mae=metrics.mean_absolute_error(Y_test,Test_data_prediction)
print("Mean Absolute Error = ",Mae)
```

Package Used

1. **NumPy**: NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
2. **Pandas**: Pandas is a software library written for the Python programming language for data manipulation and analysis. It offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.
3. Scikit-learn: Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k*-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
4. Matplotlib : Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

   One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

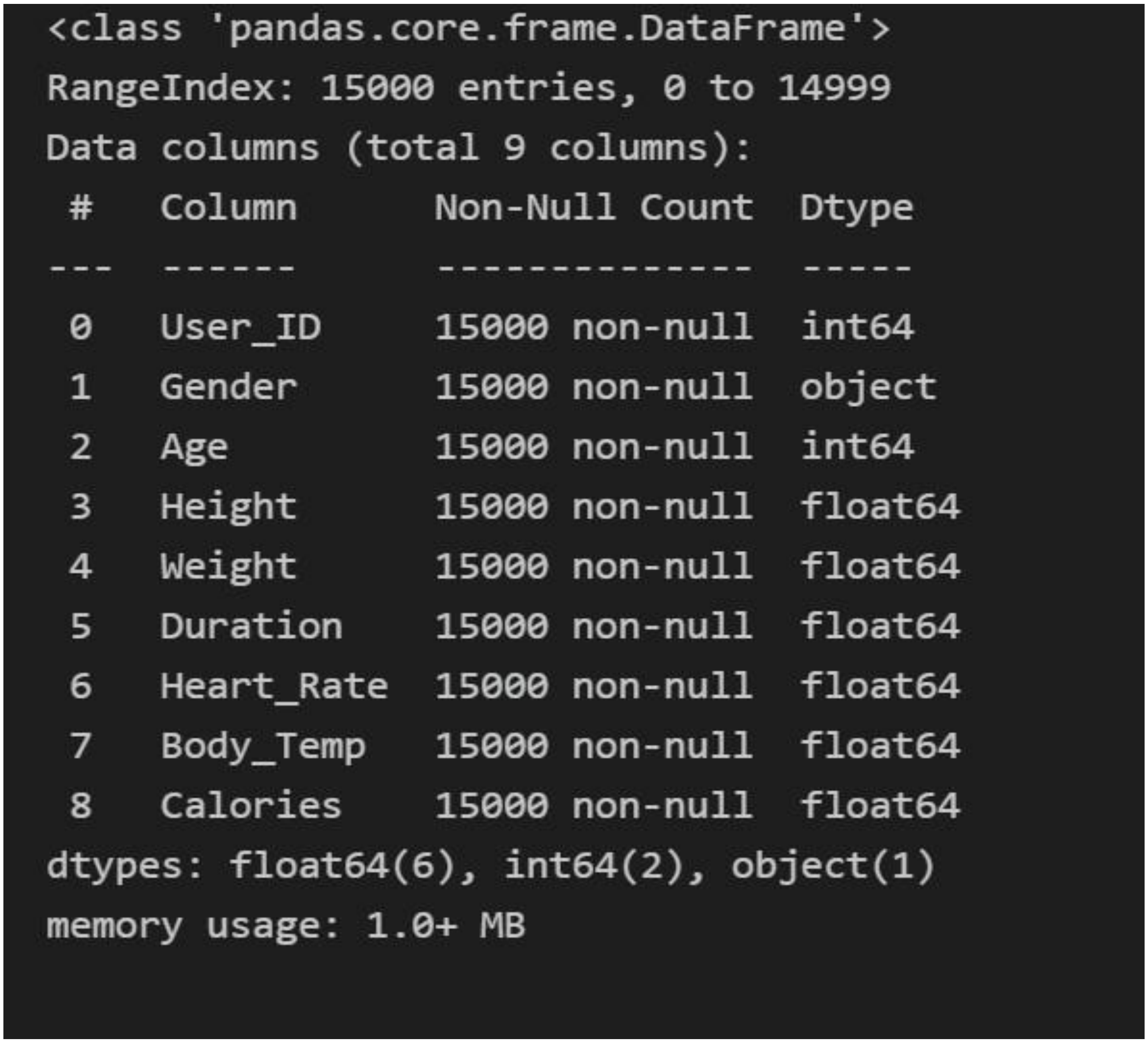
   Matplotlib comes with a wide variety of plots. Plots helps to understand trends, patterns, and to make correlations. They're typically instruments for reasoning about quantitative information. Some of the sample plots are covered here.

5. Seaborn: Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.

   Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

Model Description

The Exercise dataset extricated from the UCI machine learning store is utilized for usage. The dataset comprises of 15,000 person information with 8 autonomous highlights (User_ID, Gender, Age, Height, Weight, Duration, Heart Rate, Body Temperature) and 1 Target “Calories”. The code is implemented with python under Anaconda Navigator with Spyder IDE. The data set is splitted with 80:20 for training and testing dataset.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   User_ID     15000 non-null  int64
 1   Gender      15000 non-null  object
 2   Age         15000 non-null  int64
 3   Height      15000 non-null  float64
 4   Weight      15000 non-null  float64
 5   Duration    15000 non-null  float64
 6   Heart_Rate  15000 non-null  float64
 7   Body_Temp   15000 non-null  float64
 8   Calories    15000 non-null  float64
dtypes: float64(6), int64(2), object(1)
memory usage: 1.0+ MB
```

The Exercise dataset with 8 independent variables and 1 dependent variable has been used for implementation. The prediction of burnt out calories is done with the following contributions.

(i) Firstly, the data set is preprocessed with Feature Scaling and missing values.

(ii) Secondly, exploratory feature examination is done and the scattering of target highlight is visualized.

(iii) Replace values of gender feature by 0 & 1 for male and female respectively.

(iv) The train-test split procedure is appropriate when you have a very large dataset, a costly model to train, or require a good estimate of model performance quickly.

(v) Train data by XGBOOST Regressor Model.

(vi) Compare the test data by prediction value.

(v) Calculate Mean Absolute Error

References

1. Lili Pan, Samira Pouyanfar, Hao Chen, Jiaohua Qin, Shu-Ching Chen (2017). DeepFood: Automatic Multi-Class Classification of Food Ingredients Using Deep Learning - IEEE 3rd International Conference on Collaboration and Internet Computing, 2017, DOI 10.1109/CIC.2017.00033
2. [2] Wu W. and Yang J. (2009), “Fast food recognition from videos of eating for calorie estimation,” in Multimedia and Expo, ICME 2009. IEEE International Conference on. IEEE, (pp.1210–1213
3. [3] Pouladzadeh P., Shirmohammadi S., Bakirov A., Bulut A., & Yassine A. (2014).Cloud-based SVM for food categorization, Multimedia Tools and Applications, 74(14), 5243–5260, DOI 10.1007/s11042-014-2116-x
4. [4] Ankita Podutwar A., Pragati Pawar2 D., Abhijeet Shinde V.,(2017),A Food Recognition System for Calorie Measurement, International Journal of Advanced Research in Computer and Communication EngineeringVol. 6, Issue 1, January 2017 DOI 10.17148/IJARCCE.2017.6146
5. [5] Zhang W., Zhao D., Gong W., Li Z., Lu Q., & Yang S. (2015), Food Image Recognition with Convolutional Neural Networks. 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications

and Its Associated Workshops (UIC-ATC-ScalCom),DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.139

6. [7] Deepika Jaswal, Sowmya .V, Soman K.P. (2014), Image Classification Using Convolutional Neural Networks, International Journal of Advancements in Research & Technology, Volume 3, Issue 6, June-2014 (Pg 1661 - 1668)
7. [13] National Food Security Mission, Operational Guidelines, Department of Agriculture and Cooperation, Ministry of Agriculture, Govt. of India, 2007
8. [11] T. Longvah R. Ananthan K. Bhaskarachary K. Venkaiah “Indian Food Composition Tables”

Project link

https://github.com/prabhat1164/Calories-Burnt-Prediction-using-Machine-Learning-