



Name: C V S K Soumya

Reg.No: BL.EN.U4ECE19028

Branch: Electronics and communication
engineering

Sem: VI

Section: A

Course Code: 19ECE363

Course Name: Machine Learning

Calories Burnt prediction

Objectives:

To predict the calories burnt during the exercise using parameters like duration, average heartbeat, body temperature, height and weight.

Introduction:

Calorie measures the amount of heat or energy that is released when you burn food down. Calories are burnt through a complicated process that oxidizes food to its basic components. so the muscles need more oxygen and this oxygen is provided by the blood which in turn is pumped by the heart. since more oxygen is required the heart rate increases and when a person works out only a part of the energy is used and the rest is excreted as sweat. Therefore, the parameters that we choose are duration, average heartbeat, body temperature, height and weight.

Methodology:

The algorithm that has been used is the supervised learning algorithm's regression model.

Data set has been collected below is the sample:

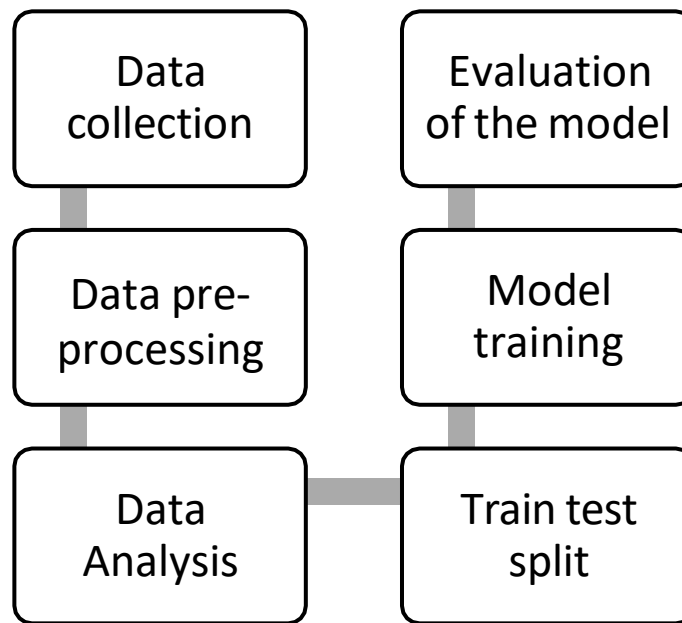
	A	B
1	User_ID	Calories
2	14733363	231
3	14861698	66
4	11179863	26
5	16180408	71
6	17771927	35
7	15130815	123
8	19602372	112
9	11117088	143
10	12132339	134
11	17964668	72
12	13723164	3
13	13681290	92

	A	B	C	D	E	F	G	H
1	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rat	Body_Temp
2	14733363	male	68	190	94	29	105	40.8
3	14861698	female	20	166	60	14	94	40.3
4	11179863	male	69	179	79	5	88	38.7
5	16180408	female	34	179	71	13	100	40.5
6	17771927	female	27	154	58	10	81	39.8
7	15130815	female	36	151	50	23	96	40.7
8	19602372	female	33	158	56	22	95	40.5
9	11117088	male	41	175	85	25	100	40.7
10	12132339	male	60	186	94	21	97	40.4
11	17964668	female	26	146	51	16	90	40.2
12	13723164	female	36	177	76	1	74	37.8
13	13681290	female	21	157	56	17	100	40

Tools being used are Jupyter notebook and VS code.

Language being used is python programming language

Flowchart of work:



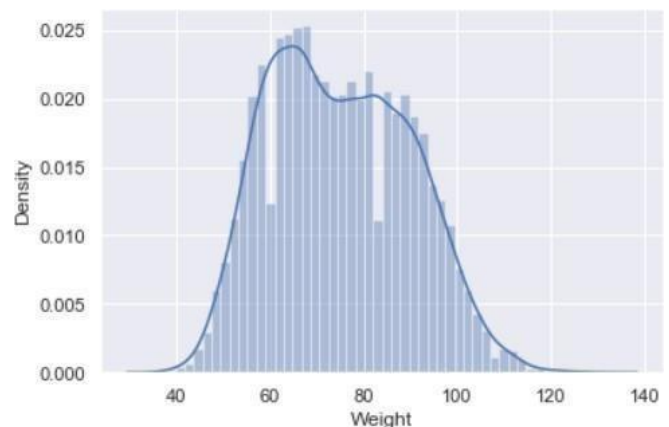
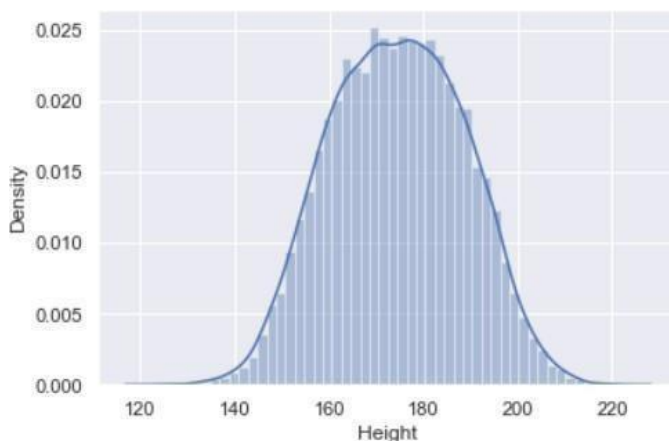
Results:

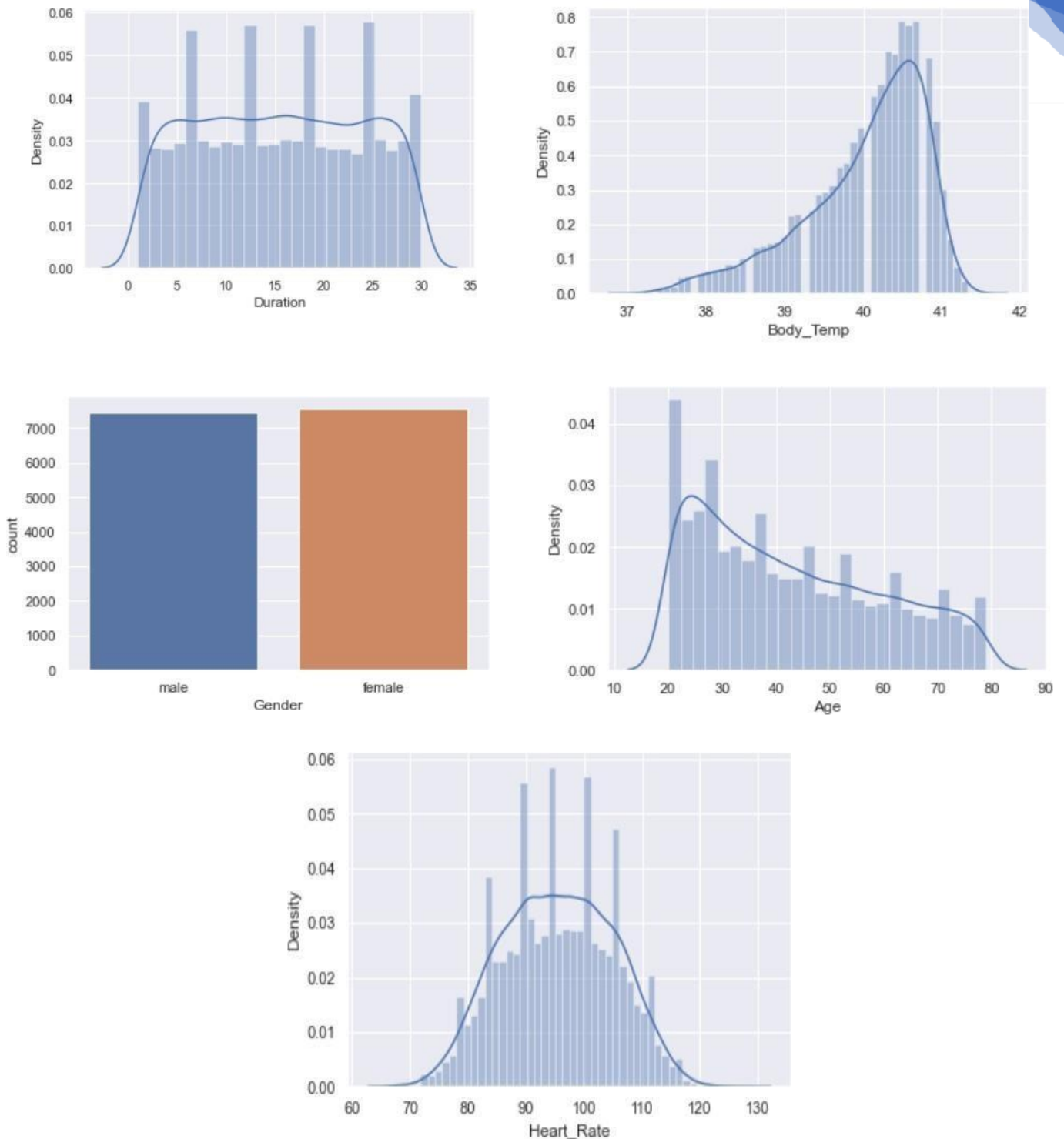
✓ Data sets:

	User_ID	Gender	Age	Height	Weight	Duration	Heart_Rate	Body_Temp	Calories
0	14733363	male	68	190.0	94.0	29.0	105.0	40.8	231.0
1	14861698	female	20	166.0	60.0	14.0	94.0	40.3	66.0
2	11179863	male	69	179.0	79.0	5.0	88.0	38.7	26.0
3	16180408	female	34	179.0	71.0	13.0	100.0	40.5	71.0
4	17771927	female	27	154.0	58.0	10.0	81.0	39.8	35.0

✓ Size of the data: (15000, 9)

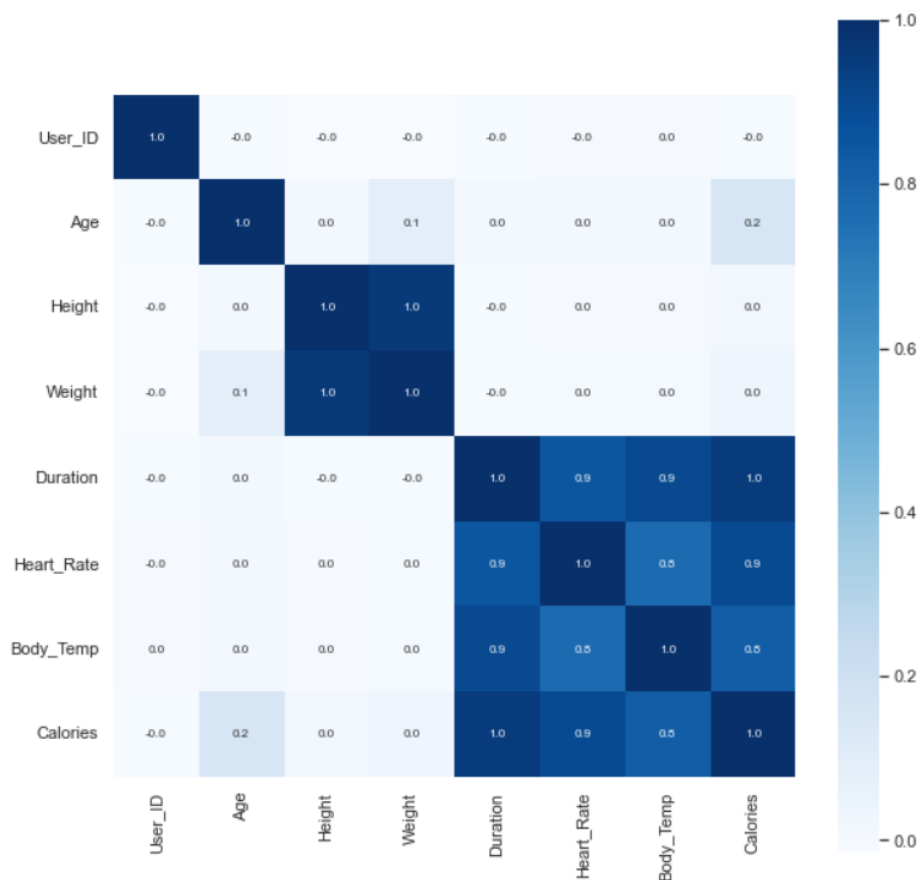
✓ Plots:





- ✓ Size of the training and testing data: (12000, 7) (3000, 7)
- ✓ Predicted output :
`[127.823784 226.00154 38.66253 ... 144.3636 22.767195 89.87375]`
- ✓ Mean absolute error: Mean Absolute Error = 1.4807048829992613

✓ Heatmap for correlation:



Summary:

To attain the output of the objective the required dependencies are imported as pandas, numpy, matplotlib etc. Loaded data from csv files namely calories.csv and exercise.csv. Combined both the csv files based on user_id. Then the dataset has been checked for missing values. Later the plots of the features in the data had been plotted and a heatmap for correlation has been constructed. Features and targets have been selected from the data set. The whole data set will be divided into four arrays x_train, y_train, x_test, y_test. x_train and x_test will be the containing the features selected. Y_train and y_test will be containing the target. The model has been trained using x_train and y_train. Further the model is evaluated using x_test and the mean absolute error between the y_test array and predicted will be calculated.

Code:

```
#importing dependencies
import os
import numpy as np #for array based calculations
import pandas as pd #for data analysis
import matplotlib.pyplot as plt #for the plotting of graphs
import seaborn as sns #for making the visualization mych attractive
from sklearn.model_selection import train_test_split #for using the supervised and
from xgboost import XGBRegressor
from sklearn import metrics

#loading data from csv file
calories=pd.read_csv('calories.csv')

#printing first five elements of the dataset
calories.head()

#loading data from csv file
exercise=pd.read_csv('exercise.csv')

#printing first five elements of the dataset
exercise.head()
calories_data=pd.merge(exercise,calories,on='User_ID')
calories_data.head()

#checking the no of rows and columns
calories_data.shape

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

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

#data analysis
#stastical measures about data
calories_data.describe()

#data visualization
sns.set()

#plotting the gender column
#countplot for categorical plot
sns.countplot(calories_data['Gender'])

#finding the dstribution of age column
sns.distplot(calories_data['Age'])

#finding the dstribution of Height column
sns.distplot(calories_data['Height'])

#finding the dstribution of weight column
sns.distplot(calories_data['Weight'])
```

```

#finding the dtribution of duration column
sns.distplot(calories_data['Duration'])

#finding the dtribution of body temperature column
sns.distplot(calories_data['Body_Temp'])

#finding the dtribution of Heart rate column
sns.distplot(calories_data['Heart_Rate'])
corelation=calories_data.corr()

#constructing a heatmap to understand the corelation
plt.figure(figsize=(10,10))
sns.heatmap(corelation,cbar=True,square=True,fmt='.1f',annot=True,
annot_kws={'size':8},cmap='Blues')

#converting the text data into the numerical data
#categorical data encoding
calories_data.replace({"Gender":{"male":0,'female':1}}, inplace=True)
calories_data.head()

#separating features and target
x=calories_data.drop(columns=['User_ID','Calories'],axis=1)
y=calories_data['Calories']
print(y)

#splitting the data into training and testing data
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)

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

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

#evaluating the model
test_data_prediction = model.predict(x_test)
print(test_data_prediction)

#mean absolute error
mae = metrics.mean_absolute_error(y_test, test_data_prediction)
print("Mean Absolute Error =",mae)

```

References:

Dataset: <https://www.kaggle.com/datasets/fmendes/fmendesdat263xdemos>