1. **INTRODUCTION**

The overarching idea of this machine learning project is to predict the calories burn during the workout. In this project we first build a machine learning system that can predict the amount of calorie burnt during exercise. In today’s world many people are inquisitive about the workout that they do and the weight loss plan that they take and how much calorie do they burn once they workout. To solve this problem, we can use ML algorithm such as XGBoost regressor.

The dataset that I am using is Exercise and Calorie dataset. Exercise dataset has 15000 entities of different persons. It has the features User Id, Age, Gender, Height, Weight, Duration of workout, average Heartbeat and Body temperature. Calorie dataset consists of Burned calories of each person. It has features User Id and Calorie. The proposed architectures in the paper are Linear Regressor and XGBoost regressor. But after a brief study, I felt that the XGBoost regressor is more useful and advantageable to this task.

1. **SUPPORTING LITERATURE**

**2.1 Literature Review**

**Paper 1.** M. Nipas, A. G. Acoba, J. N. Mindoro, M. A. F. Malbog, J. A. B. Susa and J. S. Gulmatico, "Burned Calories Prediction using Supervised Machine Learning: Regression Algorithm," 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T), 2022, pp. 1-4, doi: 10.1109/ICPC2T53885.2022.9776710.

Regular physical activities are essential to staying healthy and fit. The estimation of calories burned by individuals is based on a formula and MET charts. This study aims to predict the calories burned using a regression model as one of the machine learning algorithms to give more accurate results. Data preparation, cleaning, and analysis are the primary steps before they can be fed to the regression models. Model training and testing using K-fold validation were done to determine the best model for the study. The performance and prediction accuracy of regression models were evaluated based on the result of model testing after ten (10) iterations. The average accuracy was computed and the result shows that Random Forest regression is the best model for the study with an accuracy of 95.77%. It is very important to visualize and study the relationships of the variables in the data because it may affect the performance of the algorithm in predicting the value of the target variable. The Random Forest regression model was able to predict the calories burned with a high accuracy rate.

**Paper 2.** Sona P Vinoy “Calorie Burn Prediction Analysis Using XGBoost Regressor and Linear Regression Algorithms” Proceedings of National Conference on Emerging Computer Applications 2022(NCECA 2022), Amal Jyothi College of Engineering Kanjirappally, Kottayam, India, pp. 187-191 (978-93-5607-317-3).

The overarching idea of this research project is to make a comparative study of machine learning algorithms to predict the calories burn during the workout. In this paper we first build a machine learning system that can predict the amount of calories burnt during exercise. In today’s world many people are inquisitive about the workout that they do and the weight loss plan that they take and how much calorie do they burn once they workout. To solve this problem, we can use ML algorithms such as XGBoost regressor and Linear Regression.

From the analysis we met with a conclusion that the XGB Regressor has more accurate results than the Linear regression model. Mean absolute error imply absolute error ought to be as low as viable. it is not anything but the difference between the actual and predicted values through the models. The mean absolute error value that is getting in XGB Regressor is 2.71 which is a good value. The error values are very less. Therefore, we can conclude that the best model for the calorie burn prediction is XGBoost Regressor.

**Paper 3.** Rachit Kumar Singh, Vaibhav Gupta “Calories Burnt Prediction Using Machine Learning” International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211 Volume 9, Issue 12, December-2021, Impact Factor: 7.429

Machine Learning is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization’s target. The object of this research paper is to create a project that can be used predict calories burnt using Machine Learning with Python. Xgboost Regression model is used in this project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Year** | **Authors** | **Title** | **Model** | **Findings** |
| 1 | **2022** | M. Nipas, A. G. Acoba, J. N. Mindoro, M. A. F. Malbog, J. A. B. Susa and J. S. Gulmatico | Burned Calories Prediction using Supervised Machine Learning: Regression Algorithm | K-fold validation,  Random Forest regression | The Random Forest regression model was able to predict the calories burned with a high accuracy rate. |
| **2** | **2022** | Sona P Vinoy | Calorie Burn Prediction Analysis Using XGBoost Regressor and Linear Regression Algorithms | XGBoost regressor and Linear Regression | This is implemented using neural networks such as Convolutional Neural Network (CNN) and image pre-processing techniques. |
| **3** | **2021** | Rachit Kumar Singh, Vaibhav Gupta | Calories Burnt Prediction Using Machine Learning | XGBoost regressor | Multiple instance parameters and various factors can be used to make the calories prediction more innovative and  successful. flexibility of the proposed approach can be increased with variants. . |

Table 1.0

**2.2 Findings and Proposals**

From the literature analysis we met with a conclusion that the XGB Regressor has more accurate results than the Linear regression model. Mean absolute error imply absolute error ought to be as low as viable. it is not anything but the difference between the actual and predicted values through the models. In machine learning the XGBoost algorithm performs well since it has robust handling of many varieties of data types, relationships, distributions, and the many hyperparameters that you can fine-tune. XGBoost regressor can be used for regression, classification for both binary and multiclass, and ranking problems. So we will be using XGBoost regressor algorithm in our model.

1. **SYSTEM ANALYSIS**

**3.1 Analysis of Dataset**

**3.1.1 About the Dataset**

We use "Kaggle" as our dataset store. A total of 15000 instances and 7 attributes of data are present throughout 2 CSV files. The "Kaggle" repository's dataset comprises information about a variety of people, including their height, weight, gender, age, workout intensity, heart rate, and body temperature. The training data is taken from the “exercise.csv” and “calories.csv” datasets. Additionally, the user id-mapped target class from the second calorie dataset comprises the calories burned by the person in exercise dataset. The information gathered is both category and numerical.

**3.1.2 Explore the Dataset**

15000 instances and 7 attributes of data are contained in two csv files (“exercise.csv” and “calorie.csv”). Each person's attributes are included in the Kaggle data collection, including their height, weight, gender, age, workout duration, heart rate, and body temperature.

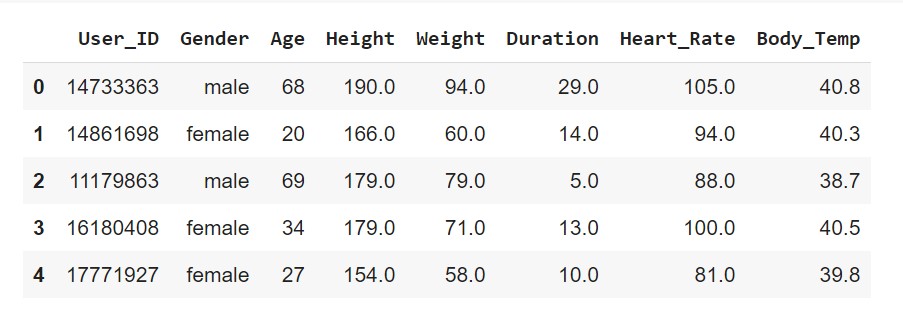




Figure 3.1

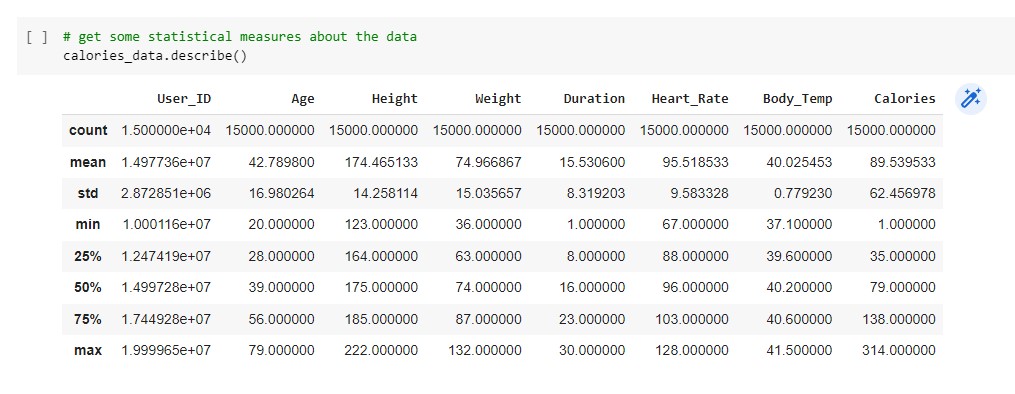


Figure 3.2

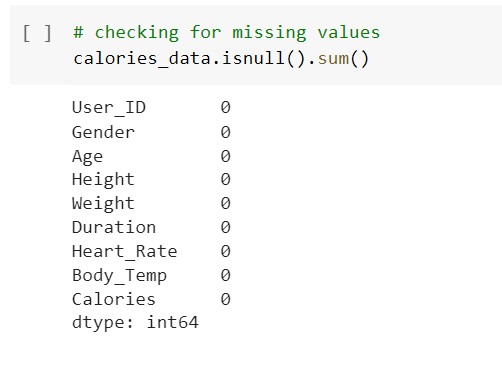
|  |  |
| --- | --- |
| **Attribute** | **Function** |
| Gender | Gender (female: 1, male: 0) |
| Age | Age is mentioned in years |
| Height | Height of the person |
| Heart\_ rate | Average heart rate of an individual during the workout (Normal heart rate 75  beats/min) |
| Body\_temp | Average body temperature captured in the course of entire workout (greater  than 37 degree Celsius) |
| Duration | Duration of exercising in minutes. |
| Calories | The total amount of calories burned while workout. |

Table 1.1

**3.2. Data Pre-processing**

**3.2.1 Data Cleaning**

Data Cleaning is the data pre-processing method we choose. Data cleaning routines attempt to fill in missing values, smooth out noisy data and correct inconsistencies. The dataset taken is already pre-processed, so pre-processing techniques are not need for the dataset. But for assurance pre-processing techniques for handling missing values and duplicated values are made.

****

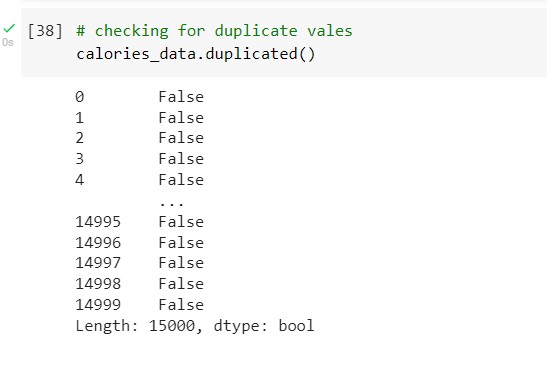
****

Figure 3.3

Label Encoding is another data pre-processing technique used for converting categorical feature list values into numerical data.



Figure 3.4

**3.2.2. Analysis of Feature Variables**

Feature variables consist of numerical data. They are Gender, Age, Height, Weight, Duration, Heart\_Rate, Body\_Temp.

**3.2.3 Analysis of Class Variables**

Class of the dataset is the category to which the input will be classified to. That means the final result of an application. In my work since there are two datasets, I have 2 classes for each dataset. The deepfake dataset has two classes real and fake. The classes are real and fake. A label of 1 is assigned to real and label of 0 is assigned to fake.

**3.3 Data Visualization**

Data visualization is the graphical representation of information and data in a pictorial or graphical format (Example: charts, graphs, and maps). Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers. Data visualization tools and technologies are essential to analysing massive amounts of information and making data-driven decisions. The concept of using pictures is to understand data that has been used for centuries. General types of data visualization are Charts, Tables, Graphs, Maps, and Dashboards.

The count of the gender is equally distributed in the dataset which can be seen in the below figure



Figure 3.5

Also, we have the mean value for age, height, and weight represented in the figure below.

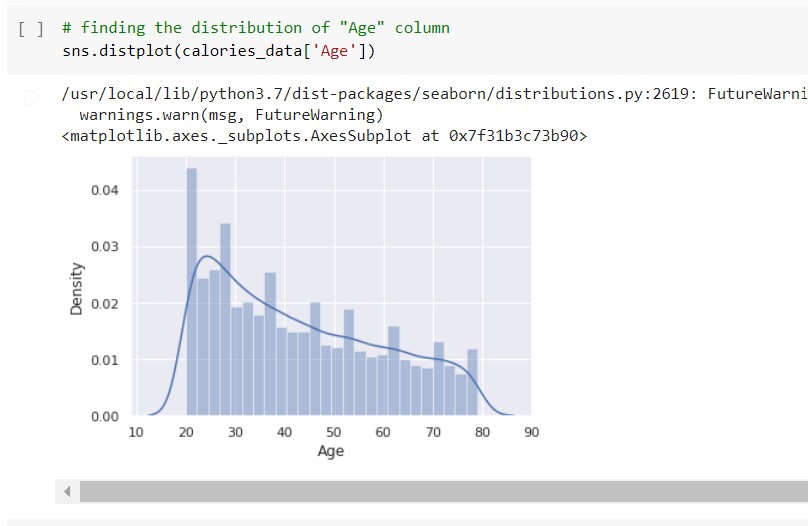


Figure 3.6

For age, more values between the age group of 20 and 30 can be seen. There is a peek in the curve means which we had generated using 15000 instances. The decrease in the curve can be seen as people tend to not workout at an older age.

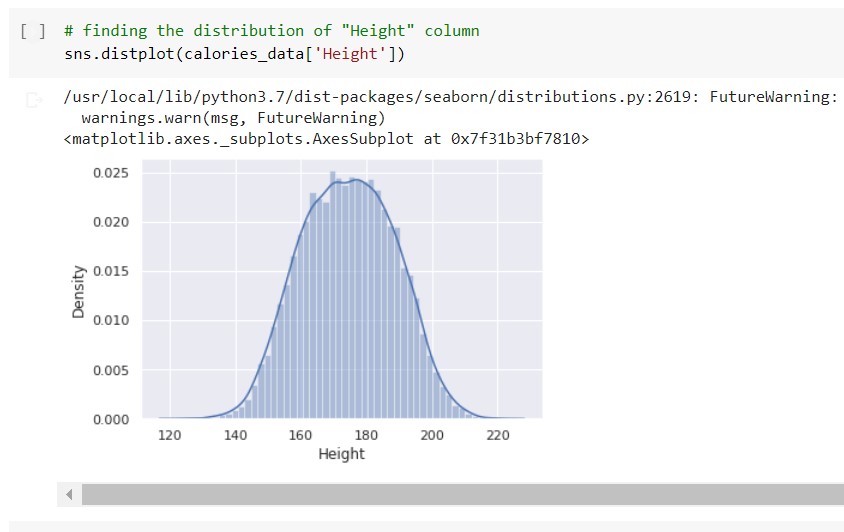


Figure 3.7

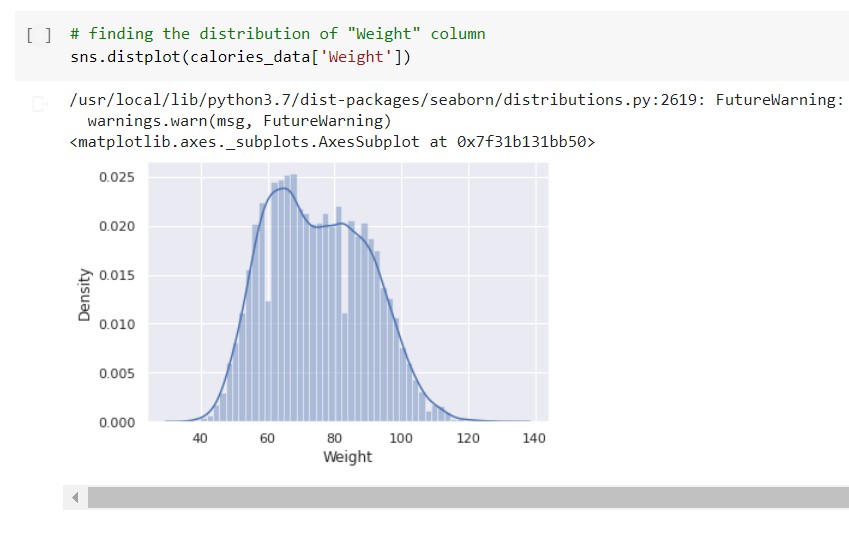


Figure 3.8

The relationship between the different records is then examined. There are two forms of correlation: positive correlation and negative correlation. The quantity of calories burned will increase as exercise duration increases. These values are therefore proportionate, i.e., in the same direction, and unquestionably connected.

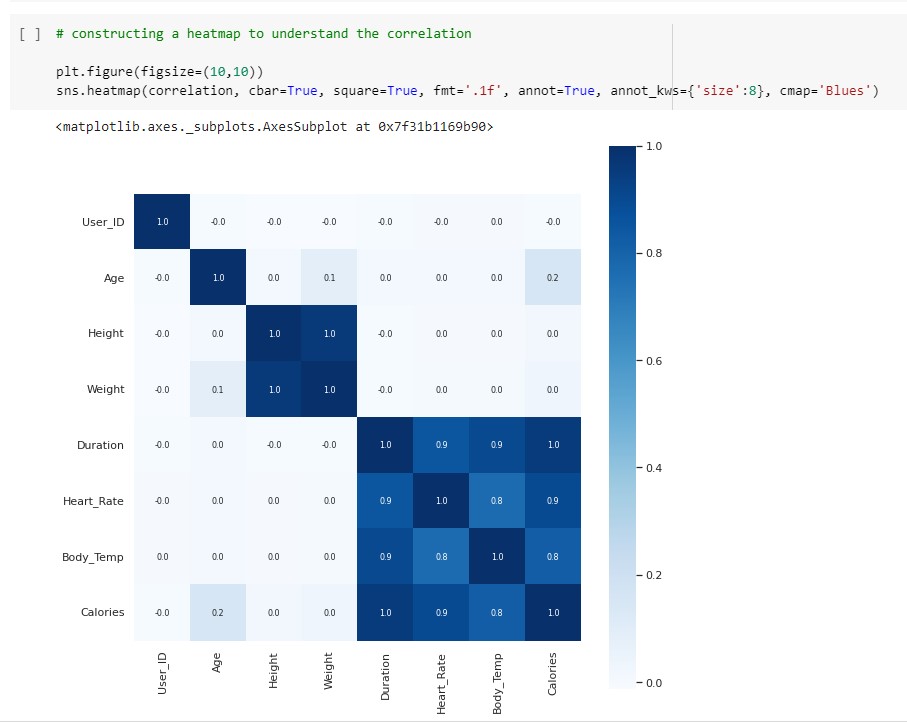


Figure 3.9

**3.4. Analysis of Algorithms**

Here we use XGBoost Regressor algorithm for prediction. XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

**XGBoost Regressor**

When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

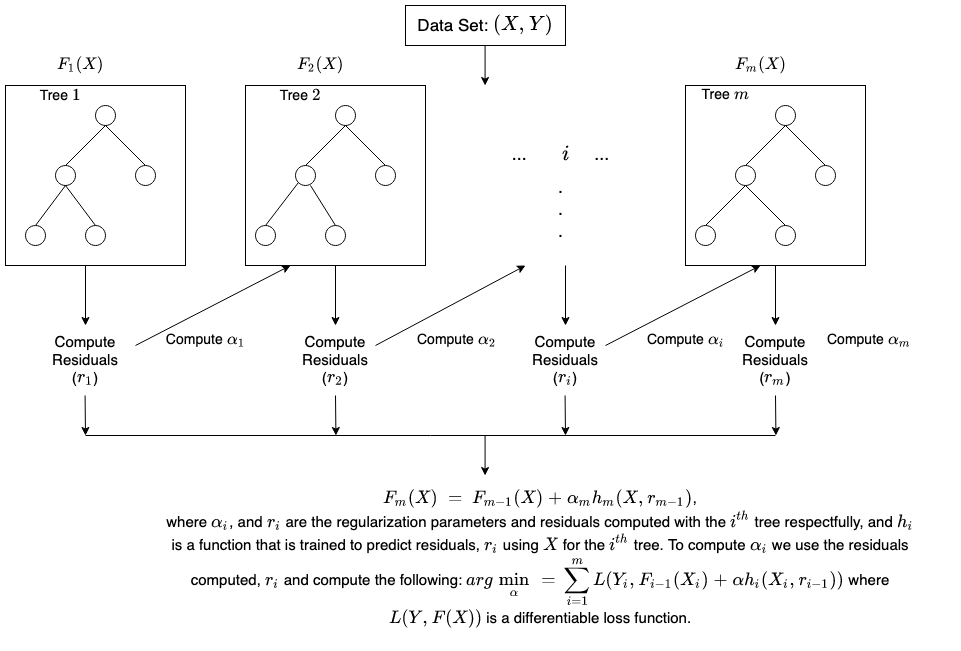


Figure 4.0

**3.5 Project Pipeline**

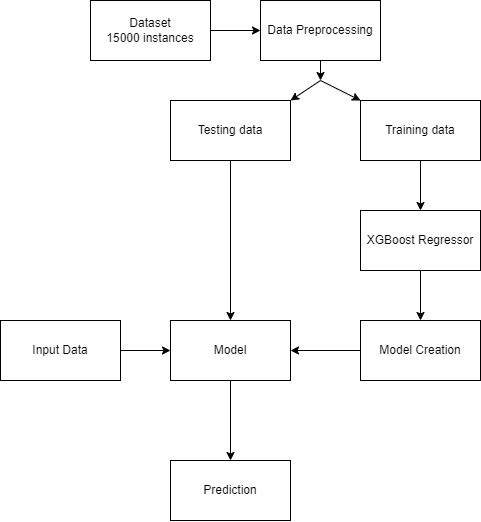
****

Figure 4.1

**Data collection**: A dataset with appropriate parameters like age, gender, height, weight, workout duration, body temperature, average heartbeat, etc. is used for the study.

**Data Pre-processing**: Make the acquired data set in an organized format. Data Cleaning is the data pre-processing method we choose. Data cleaning routines attempt to fill in missing values, smooth out noisy data and correct inconsistencies.

Missing values can be handled by:

* Ignore the tuple: This is usually done when the class label is missing.
* Use global constant to fill the missing value: Replace all missing attribute values by the same constant, such as label like “Unknown” or NA.This method is simple.
* Use attribute mean or use the attribute mean for all samples belonging to the same class as the given tuple.

Noisy data can be handled by:

* Binning: Binning methods smooth a sorted data value by consulting its “neighbourhood”. The sorted values are distributed into a number of buckets or bins. Since binning methods consult the neighbourhood of values, they perform local smoothing.
* Regression: Data can be smoothed by fitting the data to a function such as with regression. Linear regression and Multiple Linear Regression can be used.
* Clustering: Outliers may be detected by clustering, where similar values are organized into groups or clusters. Intuitively values that fall outside of the set of clusters may considered as outliers.
* The dataset taken is already pre-processed, so pre-processing techniques are not need for the dataset. But for assurance pre-processing techniques for handling missing values and duplicated values are made.

**Training and Testing**: Model Training was done for XGBoost regressor algorithm and generates model for this. The model of algorithm is saved and used for result prediction. Testing is made by loading saved model and perform prediction through python code. Accuracy comparison is made by splitting dataset into training and test data. After Testing, User interface is developed for prediction and connect the model using Flask Framework.

**3.6. Feasibility Analysis**

A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing system or proposed system, opportunities and threats present in the natural environment, the resources required to carry through, and ultimately the prospects for success.

Evaluated the feasibility of the system in terms of the following categories:

* Technical Feasibility
* Economical Feasibility
* Operational Feasibility

**3.6.1. Technical Feasibility**

Proposed system is technically feasible since all the required tools are easily available. Technical issues involved are the necessary technology existence, technical guarantees of accuracy, reliability, ease of access, data security, and aspects of future expansion. The application is technically feasible because all the technical resources required for the development and working of the application is easily available and reliable. The project is implemented in Python. Since Python supports a various libraries and packages that make the project development easier, the project was technically feasible. The codes are written in Google Colab, therefore all the libraries will be available, no need to install or import each of those. These requirements are easily available, reliable, and will make the system more time saving and require less manpower**.**

**3.6.2. Economic Feasibility**

In our proposed system ”Calorie Burned Prediction”, the development cost of the application is optimum. The system requires only a computer for working. The code is working on Google Colab. So, the colab consumes an amount of internet. The development of the system will not need a huge amount of money. It will be economically feasible.

**3.6.3. Operational Feasibility**

Operational feasibility assesses the extent to which the required system performs a series of steps to solve business problems and user requirements. Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. The developed system is completely driven and user friendly. Since the code is written on Google Colab, no need for worrying about importing or installing the libraries required. There is no need of skill for a new user to open this application and use it. The interface contains only a file upload option and a submit button. Users also need to be aware of the application initially. Then they can use it easily. So, it is feasible. But sometimes it has a GPU issues.

**3.7. System Environment**

System environment specifies the hardware and software configuration of the new system. Regardless of how the requirement phase proceeds, it ultimately ends with the software requirement specification. A good SRS contains all the system requirements to a level of detail sufficient to enable designers to design a system that satisfies those requirements. The system specified in the SRS will assist the potential users to determine if the system meets their needs or how the system must be modified to meet their needs.

**3.7.1. Software Environment**

Various software used for the development of this application are the following:

* Python

Python is a high-level programming language that lets developers work quickly and integrate systems more efficiently. This model is developed by using many of the Python libraries and packages such as:

* Pandas:

Pandas is an open-source library that is made mainly for working with relational or labelled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library.

* Matplotlib:

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. In this application, its used for plotting the graph.

* Numpy:

NumPy is a Python library used for working with arrays. NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. In this application, its used for handling arrays.

* Seaborn:

Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions. In this application, its used for plotting the graph.

* scikit-learn:

Scikit-learn (formerly scikits.learn and also known as sklearn) 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.

* Xgboost

XGBoost provides an easy-to-use scikit-learn interface for some pre-defined models including regression, classification and ranking.

* Google Colab

Colab is a free Jupyter notebook environment that runs entirely in the cloud. We can write and execute code in Python. Colab supports many machine learning libraries which can be easily loaded in the colab notebook.

* Visual Studio Code

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE

* HTML and CSS

Hyper Text Markup Language is used for creating web pages.HTML describes the structure of the web page. Here, the user interface of my project is done using HTML. Cascading Style Sheet is used with HTML to style the web pages.

* Flask

Flask is a web framework, it’s a Python module that lets you develop web applications easily. It’s having a small and easy-to-extend core: it’s a microframework that doesn’t include an ORM (Object Relational Manager) or such features. It does have many cool features like URL routing, template engine. It is a WSGI web app framework.

* Github

Git is an open-source version control system that was started by Linus Torvalds. Git is similar to other version control systems Subversion, CVS, and Mercurial to name a few. Version control systems keep these revisions straight, storing the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute. Git is the preferred version control system of most developers, since it has multiple advantages over the other systems available. It stores file changes more efficiently and ensures file integrity better.

The social networking aspect of GitHub is probably its most powerful feature, allowing projects to grow more than just about any of the other features offered. Project revisions can be discussed publicly, so a mass of experts can contribute knowledge and collaborate to advance a project forward.

**3.7.2.Hardware Environment**

Selection of hardware configuration is very important task related to the software development.

**Computer** -

Processor: 2 GHz or faster (dual-core or quad-core will be much faster)

Memory: 8 GB RAM or greater

Disk space: 40 GB or greater good internet connectivity

**4.SYSTEM DESIGN**

**4.1.Model Building**

**4.1.1.Model Planning**

This is the stage in which we apply the algorithm we've chosen (in this case, XGBoost regressor) to determine the mean absolute error. The accuracy comparison is made by using the dataset as training and testing data. By splitting the dataset, a portion is used for training the model and other for testing the model. 70% of dataset is used as training data and remaining 30% used as testing data.

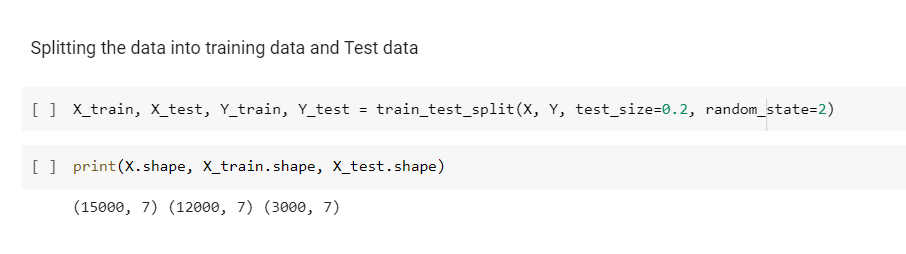


Figure 4.2

**4.1.2.Model Training**

After splitting dataset, using training data, model is generated for XGBoost algorithm.



Figure 4.3

**4.1.3. Testing**

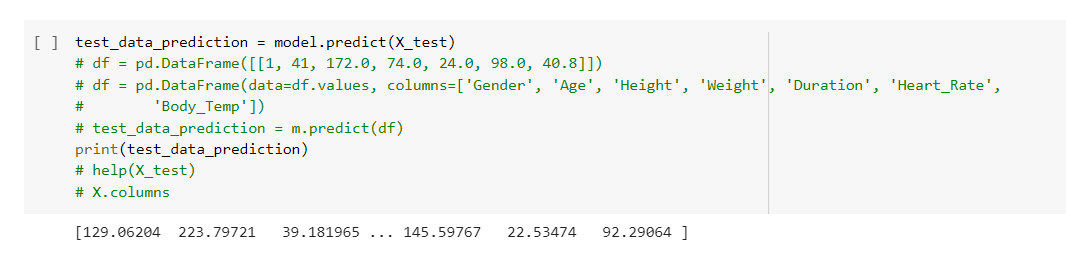
****

Figure 4.4

**5.RESULTS AND DISCUSSION**

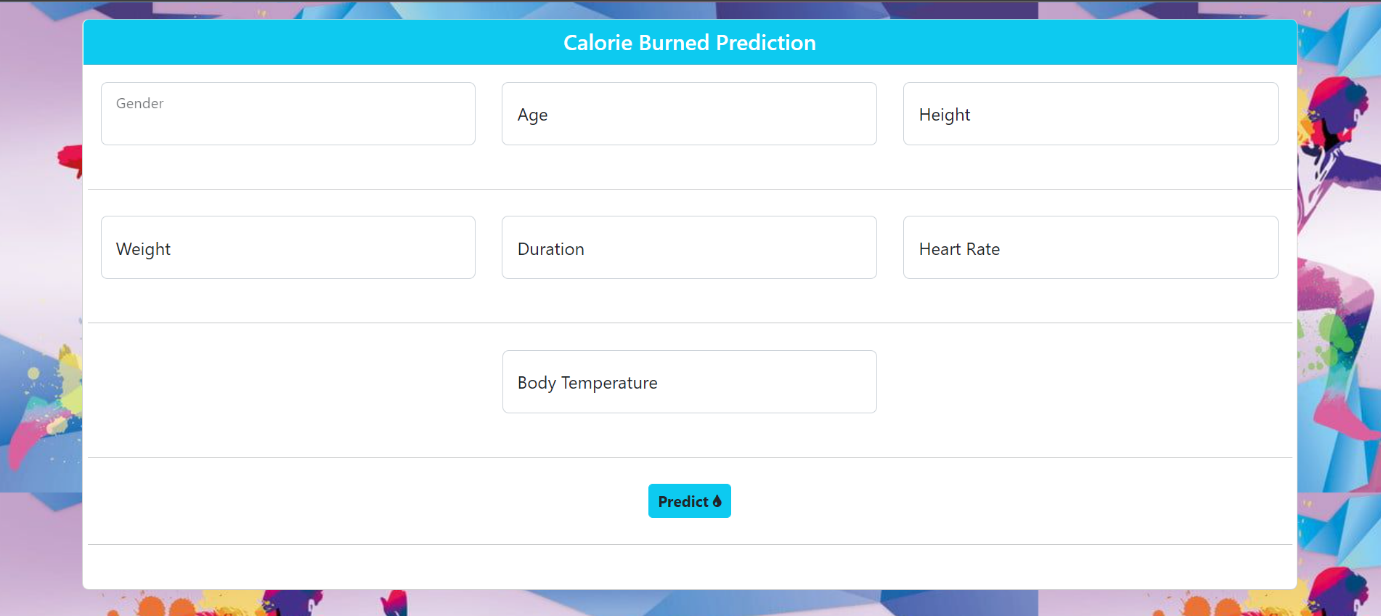
We deduced from the analysis that the XGB Regressor produces more accurate findings. Mean absolute error suggests that absolute error should be as minimal as possible. It is nothing more than the discrepancy between values that were seen and those that were predicted by models. 2.71 is a good value for the mean absolute value that the XGBRegressor gives us. The mistake rates are quite low. Therefore, we can say that XG Boost Regressor is the best model for predicting calorie burn. The flexibility of the suggested technique can also be improved with variations.

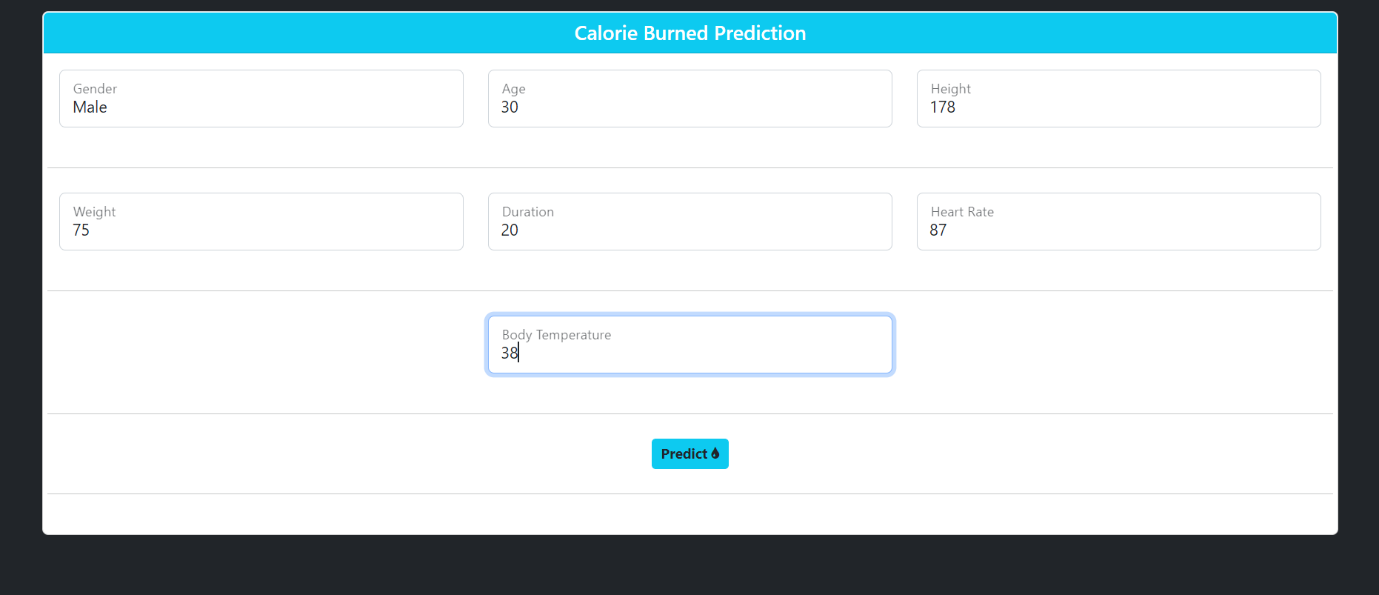


Figure 4.5

**6.MODEL DEPLOYMENT**

This figure shows the user interface of this application. The interface is very simple and easy to understand. There are 7 fields for entering details from users. There are 6 textbox fields and 1 dropdown box. There is a predict button for predict the calories. Validation for numeric field is done in html. And for making all fields mandatory validation is done when values are taken from the form to the model.





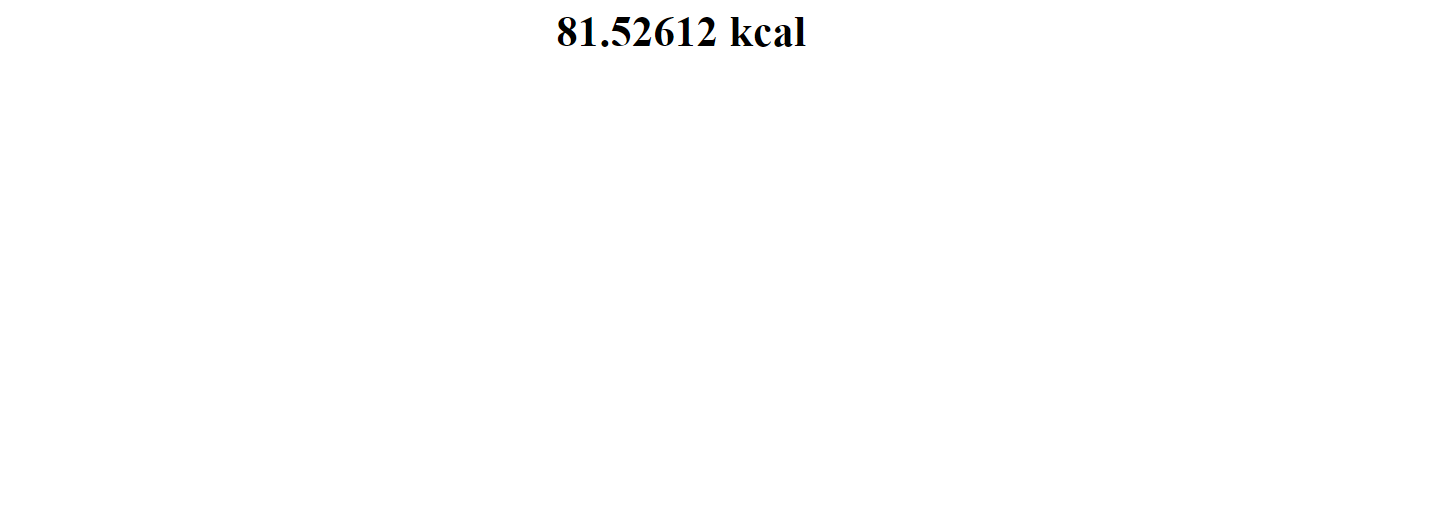


Figure 4.6

**7.GIT HISTORY**

**8.CONCLUSIONS**

This project is a machine learning project, which aims to predict how much calorie a person burns during workout. So I decided to develop a machine learning project on calorie burned prediction. For getting more understanding I referred research papers on the same topic. And each paper suggests different implementation such as Linear regression, XGBoost regressor etc. So I adapt some of the elements from each paper and implement proposed system using XGBoost which has high accuracy.

I used Google Colab for developing this application and using Visual Studio Code for designing User interface. Flask Framework is used for connecting the model with UI.

**9.FUTURE WORK**

The Calorie burned prediction model can be implemented in following sectors:

1) Fitness accessories.

2) Medical accessories.

**10.APPENDIX**

**10.1.Minimum Software Requirements**

Software: Google Colab

Operating System: Windows 10

**10.2.Minimum Hardware Requirements**

Hard disk capacity : 256 GB (minimum)

RAM : 8 GB

Processor : Intel Core i5 preferred

Display : 1366 \* 768

SSD : 512 GB

**11.REFERENCES**

* M. Nipas, A. G. Acoba, J. N. Mindoro, M. A. F. Malbog, J. A. B. Susa and J. S. Gulmatico, "Burned Calories Prediction using Supervised Machine Learning: Regression Algorithm," 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T), 2022, pp. 1-4, doi: 10.1109/ICPC2T53885.2022.9776710.
* Sona P Vinoy “Calorie Burn Prediction Analysis Using XGBoost Regressor and Linear Regression Algorithms” Proceedings of National Conference on Emerging Computer Applications 2022(NCECA 2022), Amal Jyothi College of Engineering Kanjirappally, Kottayam, India, pp. 187-191 (978-93-5607-317-3).
* Rachit Kumar Singh, Vaibhav Gupta “Calories Burnt Prediction Using Machine Learning” International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211 Volume 9, Issue 12, December-2021, Impact Factor: 7.429
* <https://www.kaggle.com/code/fmendes/exercise-and-calories>