International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211
Volume 9, Issue 12, December-2021, Impact Factor: 7.429, Available online at: www.ijaresm.com

Calories Burnt Prediction Using Machine Learning

Rachit Kumar Singh[1], Vaibhav Gupta[2]

[1]Student, Information Technology, Maharaja Agrasen Institute of Technology, Rohini, Delhi
[2]Student, Information Technology, Maharaja Agrasen Institute of Technology, Rohini, Delhi

---*****************---

ABSTRACT

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.

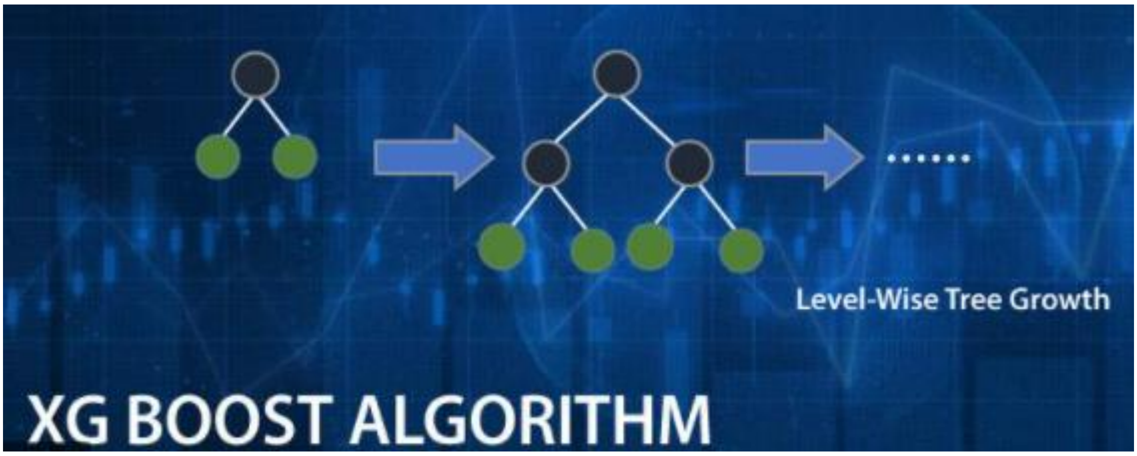
INTRODUCTION

The carbohydrates are broken into glucose and converted into energy using oxygen. The muscles that are doing exercise need more oxygen and as the body requires more oxygen the heart beat will increase a lot, so increased heart beat means increased blood flow which in turn will give more oxygen to the muscle which is used to break those glucose molecules , so the energy from these glucose molecules is used and when it is used only a part is used and rest is converted into heat , so body temperature would increase and our body will sweat, so the parameters which we would be taking into consideration for input are :- Duration for which the exercise is done, Average heart beat per minute, Body temperature, Height , weight and gender of the person .

All these would be used to create a prediction model also for calories burnt.

TECHNOLOGY USED

Xgboost Regressor:



Xgboost regressor has two parameters lambda(λ) which denotes Regulation parameter , more the regulator parameter more is the tuning of the decision tree and the other parameter is gamma(γ) which is threshold.

If Similarity Weight(SM) =Σ(Residue)^2/(No of Residues + λ)
Than Gain = Similarity Weight(left decision tree) + Similarity Weight(right decision tree) –

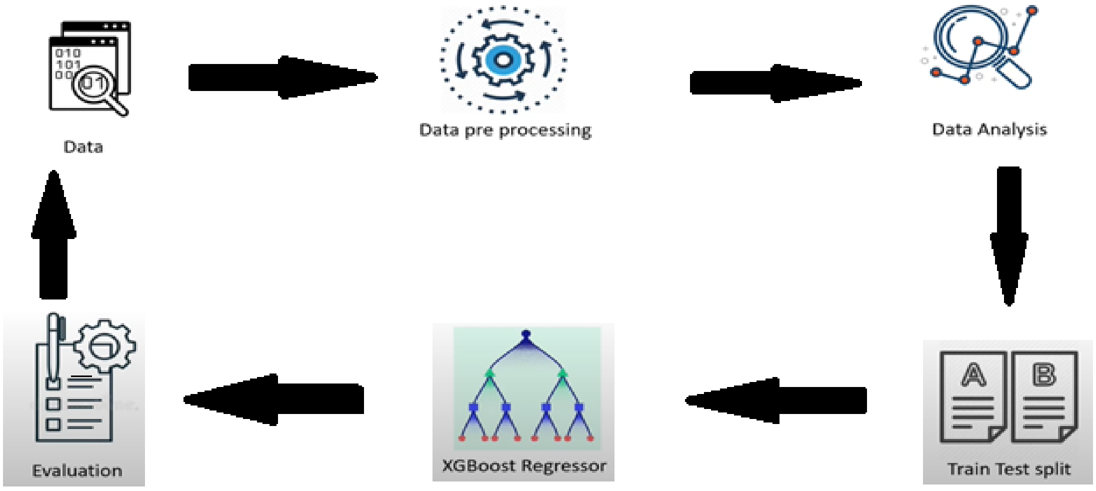
Similarity Weight(root).
If(Gain > γ) than decision tree bifurcation takes place for further levels else not takes place, this makes the xgboost algorithm efficient as compared to others.

Following are the features of xgboost

- It is the most famous algorithm of xgboost.
- Tianqi-Chan was the founder of xgboost.
- It is platform free.
- It is integrable with multiple systems.
- Xgboost has high speed of processing.
- Xgboost uses parallelization, uses maximum available computational power of the system.
- Xgboost keeps all intermediate calculations in cache so that we don't have to do the same calculation again and again.
- If we have data such that size of our data is more than the size of the memory than xgboost optimized data can work on data greater than the size of the RAM.
- Xgboost carry autotuning of decision tree with the help of regulation parameters that are lambda, gamma and eta

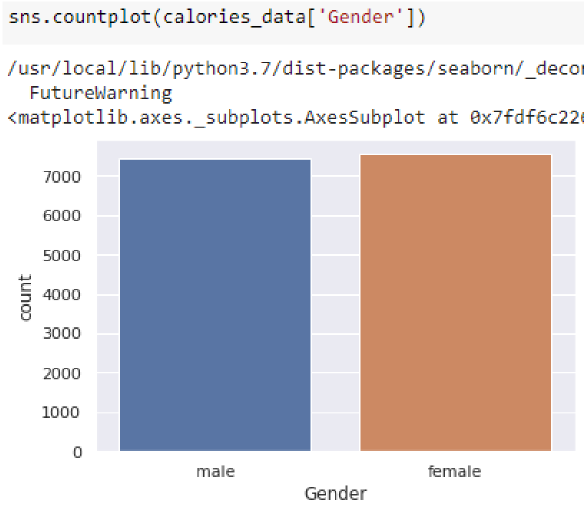
METHODOLOGY

- The project starts with importing numpy, pandas, matplotlib, pylot, sns ,xgboost regressor and other libraries.
- The data is analysed that the heart rate and body temperature would be more when the person is doing exercise.
- The data is than visualized using distribution graphs and sns library is used to give grid lines during plot. We will find the distribution of density v/s age, distribution of density v/s height and many more.
- Than we will find the correlation in the dataset by constructing heat map.
- Than we will train the model using xgboost regressor .
- Evaluate the model based on test data and compare the data with original value and find the mean absolute error.
- Than we will make a prediction model on training data , built a predictive system based on which we can predict output from a single input.



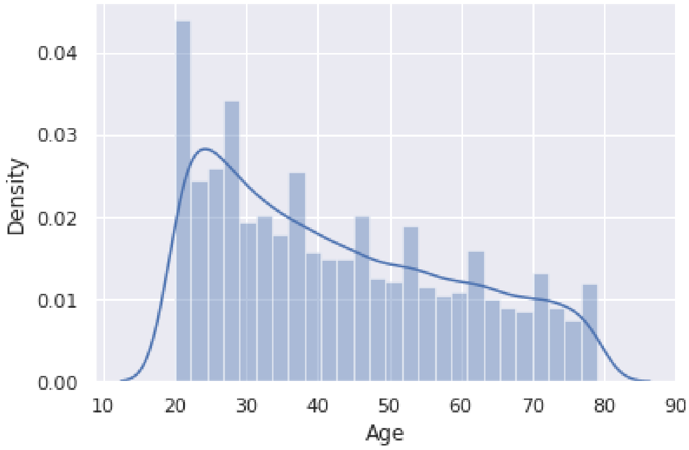
RESULTS

Data Visualization **:-** find distribution of gender column for example how many males in this data point and how many females in this data point in count plot

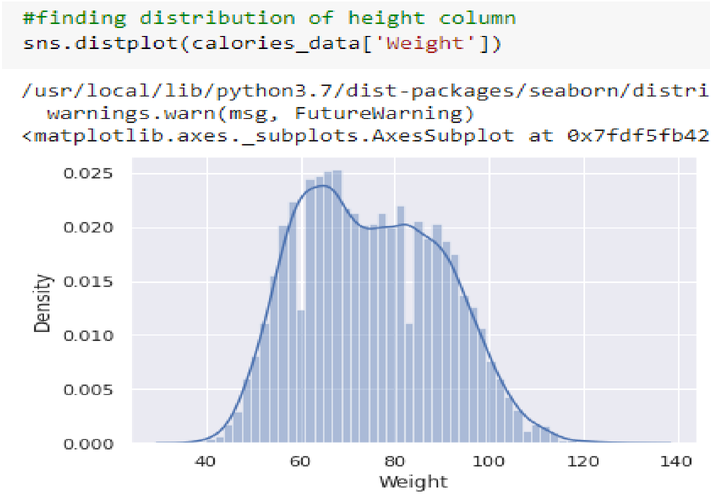


```
sns.countplot(calories_data['Gender'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decor
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fdf6c22(
```



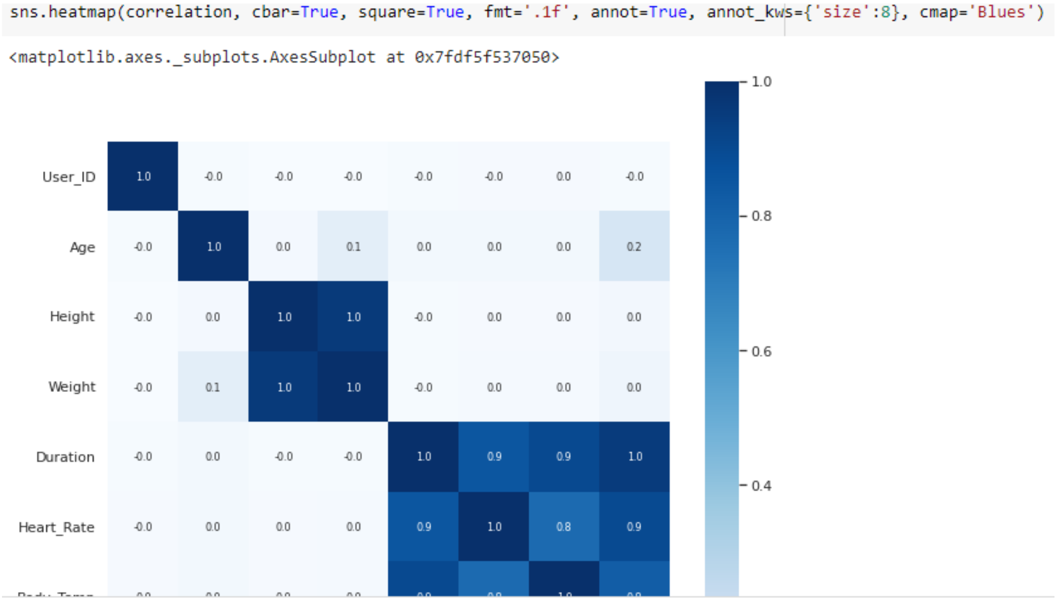
finding distribution of height column



```
#finding distribution of height column
sns.distplot(calories_data['Weight'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distri
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fdf5fb42
```

Constructing a heat map to understand the correlation.
Heat map gives colours based on values, and these values are calculated based on the relationship between the data.
Each column would be compared to the other column and if the value is 1 than two columns are positively correlated if the value is 0 than two columns are not correlated, if less than 0 than two columns are negatively correlated.
So from graph it can be seen duration, heart rate and body temperature are positively related to calories burnt.
This is how we will find the calories burnt.

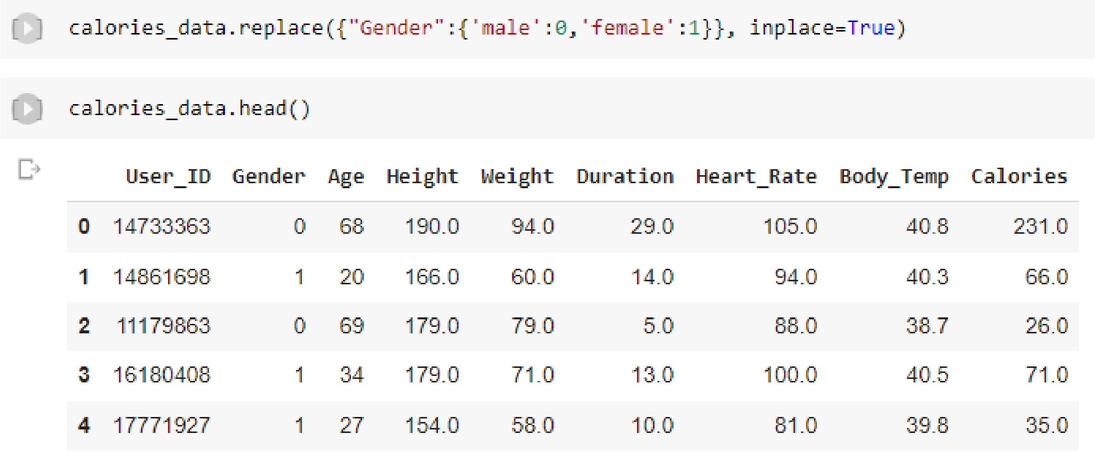


```
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot=True, annot_kws={'size':8}, cmap='Blues')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fdf5f537050>
```

User_ID	1.0	-0.0	-0.0	-0.0	-0.0	-0.0	0.0	-0.0
Age	-0.0	1.0	0.0	0.1	0.0	0.0	0.0	0.2
Height	-0.0	0.0	1.0	1.0	-0.0	0.0	0.0	0.0
Weight	-0.0	0.1	1.0	1.0	-0.0	0.0	0.0	0.0
Duration	-0.0	0.0	-0.0	-0.0	1.0	0.9	0.9	1.0
Heart_Rate	-0.0	0.0	0.0	0.0	0.9	1.0	0.8	0.9

Now all the values in our data frame are in the form of numerical values but gender column is in the form of text so we convert male to 0 and female to 1, so that our machine learning model understands better.
Converting the text data to numerical values.

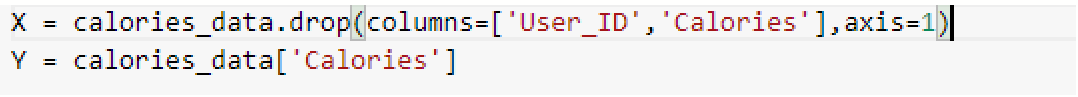


```
calories_data.replace({"Gender":{'male':0,'female':1}}, inplace=True)
```

```
calories_data.head()
```

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

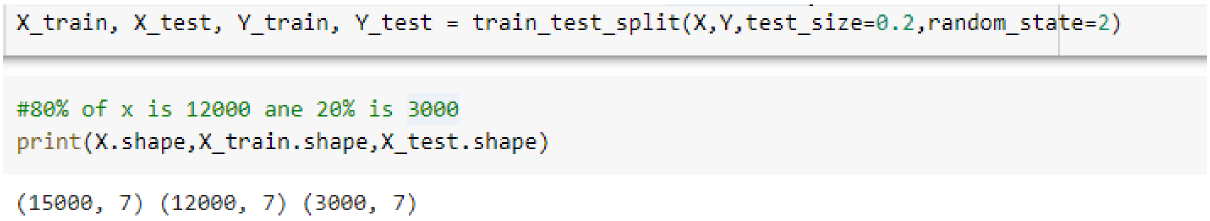
separate features and target (calories column from other columns) by drop x won't contain usesrid and calories column.



```
X = calories_data.drop(columns=['User_ID','Calories'],axis=1)
Y = calories_data['Calories']
```

Splitting the data into training data and test data xtrain represents all the training data and xtest represents all the test data (x split into two) the corresponding calories values for xtrain goes to ytrain and corresponding calories values for xtest goes to ytest.

I want 20% of the data as test data and 80% of the data as training data, my data will be split in a specific manner by random state.

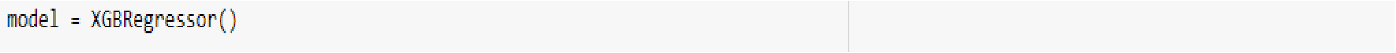


```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.2,random_state=2)
```

```
#80% of x is 12000 ane 20% is 3000
print(X.shape,X_train.shape,X_test.shape)
```

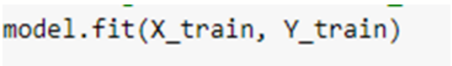
```
(15000, 7) (12000, 7) (3000, 7)
```

Model training (we are going to train xgboost regressor model), loading the model (inside the model we are loading the xgbregressor)



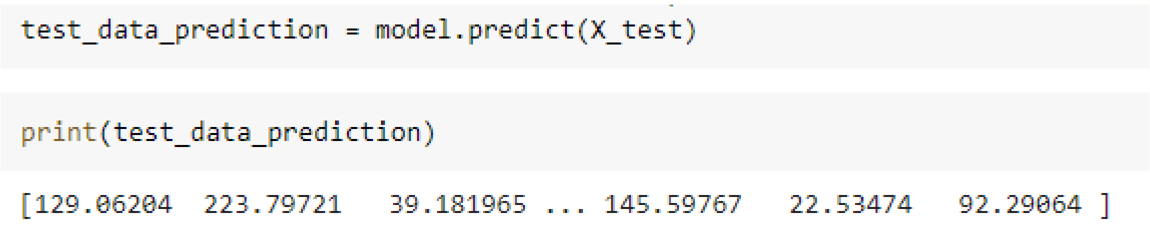
```
model = XGBRegressor()
```

Training the model with X_train, my xgboost regressor will learn from the model automatically.



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

Evaluation our model based on our test data (prediction on test data),now we will give xtest and we won't give ytest which is the calories burnt, so with this test data our model can find what is the calories burnt and we are going to compare that value with the original value which is ytest.
Now what happens is our model goes through this xtest and finds the calories burnt values for this xtest and all those values would be stored in this test data prediction.

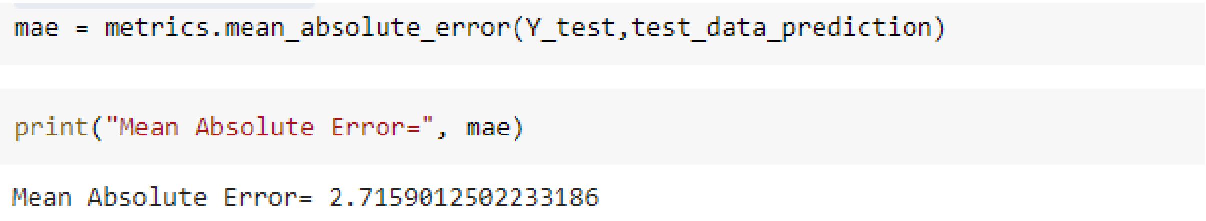


```
test_data_prediction = model.predict(X_test)
```

```
print(test_data_prediction)
```

```
[129.06204  223.79721   39.181965 ... 145.59767   22.53474   92.29064 ]
```

Compare the values predicted by our model with the original values, the original values are the ytest
Mean Absolute Error



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

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

```
Mean Absolute Error= 2.7159012502233186
```

building the predictive system based on input
input_data = (1,0,68,190,94,29,105,40.8)

changing tuple data type above to NumPy array
input_data_as_numpy_array = np. as array(input_data)

reshape this array (we are giving 1 data point, if we dont predict it the model does not know because in training we used huge data frame and our model would be expecting same number of data frame so we don't want that, we want our model to understand that we just want value for one particular data point)
input_data_reshaped = input_data_as_numpy_array. reshape (1, -1)
prediction = XGBRegressor.predict(input_data_reshaped)

CONCLUTIONS AND FUTURE SCOPE

Multiple instance parameters and various factors can be used to make the calories prediction more innovative and successful. Accuracy, which plays a key role in prediction-based systems, can be significantly increased. One of the main highlights is more expressive regression outputs, which are more understandable bounded with some of accuracy. Moreover, the flexibility of the proposed approach can be increased with variants .

REFERENCES

[1]. Smola, A., & Vishwanathan, S. V. N. (2008). Introduction to machine learning. *Cambridge University, UK*, *32*, 34.
[2]. Saltz, J. S., & Stanton, J. M. (2017). *An introduction to data science*. Sage Publications.
[3]. Shashua, A. (2009). Introduction to machine learning: Class notes 67577. *arXiv preprint arXiv:0904.3664*.
[4]. MacKay, D. J., & Mac Kay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.
[5]. Daumé III, H. (2012). A course in machine learning. *Publisher, ciml. info*, *5*, 69.
[6]. Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.
[7]. Cerrada, M., & Aguilar, J. (2008). Reinforcement learning in system identification. In *Reinforcement Learning*. IntechOpen.
[8]. Welling, M. (2011). A first encounter with Machine Learning. *Irvine, CA.: University of California*, *12*.
[9]. Learning, M. (1994). Neural and Statistical Classification. *Editors D. Mitchie et. al*, 350.
[10]. Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM*, *42*(11), 30-36.
[11]. Downey, A. B. (2011). *Think stats*. "O'Reilly Media, Inc.".
[12]. Géron, A. (2019). *Hands-On Machine Learning with Scikit- Learn, Keras, and TensorFlow: Concepts, Tools*