NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI

CSPC 54

INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

PROJECT REPORT

SUBMITTED BY:
PRINCE CHAUDHARY-(106121095)
SANJEEV PRASAD – (106121115)
ADITYA KAMBLE –(106121009)
KUNDAN KUMAR – (106121069)
RISHABH KUMAR – (106121101)

Title: Predictive Modeling of Calories Burned using XG Boost Regression in the Context of Contemporary Lifestyle

GitHub Link

Abstract

In the fast-paced environment of modern life, characterized by hectic schedules and demanding work commitments, individuals often struggle to maintain a healthy lifestyle. Neglecting dietary habits and insufficient physical activity contribute to the rising issue of obesity. Recognizing the importance of staying fit, many people turn to diets and exercise, emphasizing the need for a comprehensive understanding of calories consumed and burned.

While tracking consumed calories is relatively straightforward with information available on product labels and the internet, estimating calories burned poses a challenge due to the limited availability of accurate measurement devices. This study addresses this gap by employing a machine learning (ML) algorithm, specifically the XG Boost Regression model, to predict calories burned.

Introduction:

In the contemporary landscape of fast-paced lifestyles and demanding work commitments, the pursuit of a healthy and balanced life has become increasingly challenging. With the prevalence of sedentary habits and irregular dietary practices, the specter of obesity looms large, presenting a significant public health concern. Recognizing the critical role of physical activity in maintaining well-being, individuals are turning to diet and exercise to counteract the effects of modern living.

A pivotal aspect of this health-conscious endeavor lies in understanding the delicate equilibrium between caloric intake and expenditure. While monitoring consumed calories has become more accessible with nutritional information available on product labels and online resources, estimating calories burned remains a complex task. The scarcity of accurate measurement devices poses a substantial challenge, prompting the need for innovative solutions.

This research project seeks to address this challenge by harnessing the power of machine learning, specifically through the application of the eXtreme Gradient Boosting (XGBoost) regression model. By leveraging a dataset comprising over 15,000 data points, this study aims to predict calories burned with a high degree of accuracy. The XGBoost algorithm, known for its efficiency in handling complex datasets and delivering precise predictions, emerges as a potent tool in unraveling the intricacies of caloric expenditure.

As we delve into the details of this study, we will explore the methodology employed, the significance of the chosen evaluation metrics, and the promising initial results obtained. The objective is not only to offer a solution to the current challenges in tracking calories burned but also to contribute valuable insights into the potential of machine learning algorithms in promoting health and well-being in the face of evolving lifestyles. In doing so, this research aspires to pave the way for future advancements in utilizing predictive modeling as a tool for personal health management in our dynamic and fast-evolving world.

Objective:

The primary objective of this study is to develop an accurate predictive model for calories burned, leveraging the XG Boost Regression algorithm. The model utilizes a dataset comprising more than 15,000 data points, aiming to provide precise and reliable predictions.

Methodology:

The study employs the XG Boost Regression model, a powerful machine learning algorithm known for its efficiency in handling complex datasets and delivering accurate predictions. The model is trained on a dataset that includes information about various factors influencing calories burned, such as activity level, duration, and individual characteristics.

The Mean Absolute Error (MAE) is chosen as the evaluation metric to assess the accuracy of the model. The initial results demonstrate a promising MAE of 2.7, indicating the model's capability to predict calories burned with a high level of accuracy.

XGBoost:

XGBoost, which stands for Extreme Gradient Boosting, is a powerful and efficient machine learning algorithm that has gained widespread popularity for its performance and effectiveness in various types of predictive modeling tasks. It belongs to the family of gradient-boosting algorithms, which are ensemble learning methods.

- XGBoost is based on the gradient boosting framework, which involves building a series of weak learners (typically decision trees) sequentially, with each one correcting the errors of the previous ones.

- XGBoost incorporates a regularization term in its objective function, which helps prevent overfitting. Regularization is a technique used to penalize complex models, promoting simpler models that generalize better to new, unseen data.

- The base learners in XGBoost are decision trees, specifically shallow trees. Shallow trees have a limited depth, which helps prevent overfitting and allows the model to generalize well.

- XGBoost has a built-in mechanism to handle missing values in the dataset. It can automatically learn the best imputation strategy during the training process.

Evaluation Metrics:

Accurate assessment of the predictive model's performance is crucial for determining its reliability and effectiveness. In this study, the Mean Absolute Error (MAE) is employed as the primary evaluation metric. MAE is a straightforward and interpretable measure that quantifies the average magnitude of errors between predicted and actual values. The formula for MAE is given by:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |Y_i - \hat{Y}_i|$$

where Y_i represents the actual calorie expenditure for the ith observation, $\hat{Y}_i$ represents the predicted calorie expenditure for the ith observation, and n is the total number of observations.

Results and Discussion:

The XG Boost Regression model has shown significant potential in predicting calories burned. The mean absolute error of 2.7 suggests a close alignment between the predicted and actual values. This level of accuracy is crucial for individuals striving to maintain a healthy lifestyle by managing their calorie intake and expenditure.

It is noteworthy that the model's performance is expected to improve over time as additional data is incorporated into the training process. This ongoing enhancement will further refine the predictive capabilities of the XG Boost Regression model, ensuring its applicability in diverse scenarios and for individuals with varying activity levels.

Conclusion:

In conclusion, the application of machine learning, specifically the XG Boost Regression model, holds promise in accurately predicting calories burned. This study contributes to addressing the challenges posed by contemporary lifestyles, where individuals face difficulties in maintaining a balance between their busy schedules and health requirements.

As the model continues to evolve with additional data, it is anticipated that its predictive accuracy will further improve. This research lays the foundation for future advancements in utilizing machine learning algorithms to promote health and well-being in the face of changing lifestyles and increasing concerns about obesity.

PROJECT CODE

```
%matplotlib inline
import numpy as np
import pandas as pd
import seaborn as sns
from ydata_profiling import ProfileReport
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, ExtraTreesClassifier
from sklearn.model_selection import train_test_split
from xgboost import XGBRegressor
from sklearn import metrics
from sklearn.metrics import r2_score
```

Data Collection¶ and Exploration

```
calories = pd.read_csv("C://Users//pc513//Machine Learning Practice//Basic Project//Calories Brunt Regression Modal//calories.csv")
```

```
calories.columns
```

```
Index(['User_ID', 'Calories'], dtype='object')
```

```
exercise_data = pd.read_csv("C://Users//pc513//Machine Learning Practice//Basic Project//Calories Brunt Regression Modal//exercise.csv")
```

```
exercise_data.columns
```

```
Index(['User_ID', 'Gender', 'Age', 'Height', 'Weight', 'Duration',
       'Heart_Rate', 'Body_Temp'],
      dtype='object')
```

```
exercise_data.head()
```

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

```
calories.head()
```

```
    User_ID  Calories
0  14733363     231.0
1  14861698      66.0
2  11179863      26.0
3  16180408      71.0
4  17771927      35.0
```

```
calories_data = pd.concat([exercise_data, calories['Calories']], axis=1)
```

```
calories_data.head(10)
```

```
   User_ID  Gender  Age  Height  Weight  Duration  Heart_Rate  Body_Temp  \
0  14733363    male   68   190.0    94.0      29.0       105.0       40.8
1  14861698  female   20   166.0    60.0      14.0        94.0       40.3
2  11179863    male   69   179.0    79.0       5.0        88.0       38.7
3  16180408  female   34   179.0    71.0      13.0       100.0       40.5
4  17771927  female   27   154.0    58.0      10.0        81.0       39.8
5  15130815  female   36   151.0    50.0      23.0        96.0       40.7
6  19602372  female   33   158.0    56.0      22.0        95.0       40.5
7  11117088    male   41   175.0    85.0      25.0       100.0       40.7
8  12132339    male   60   186.0    94.0      21.0        97.0       40.4
9  17964668  female   26   146.0    51.0      16.0        90.0       40.2

   Calories
0     231.0
1      66.0
2      26.0
3      71.0
4      35.0
5     123.0
6     112.0
7     143.0
8     134.0
9      72.0
```

```
calories_data.shape
```

```
(15000, 9)
```

```
calories_data.info()
```

```
<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
```

```
profile = ProfileReport(calories_data, title="Profiling Report")
```

```
profile.to_file("your_report.html")
```

{"model_id":"b035c7a5d33944658339814094e80c0","version_major":2,"version_minor":0}

{"model_id":"32bdc7a2e41545008ccfa9c35008b957","version_major":2,"version_minor":0}

{"model_id":"4b1538640b154b049e4c571ba0fe822e","version_major":2,"version_minor":0}

{"model_id":"881db281c4124cb29db6e40227ef12d8","version_major":2,"version_minor":0}

```
calories_data.isnull().sum()
```

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

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  \
0  14733363       0   68   190.0    94.0      29.0       105.0       40.8
1  14861698       1   20   166.0    60.0      14.0        94.0       40.3
2  11179863       0   69   179.0    79.0       5.0        88.0       38.7
3  16180408       1   34   179.0    71.0      13.0       100.0       40.5
4  17771927       1   27   154.0    58.0      10.0        81.0       39.8

   Calories
0     231.0
1      66.0
2      26.0
3      71.0
4      35.0
```

Train and Test Split

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

```
X.head()
```

```
   Gender  Age  Height  Weight  Duration  Heart_Rate  Body_Temp
0       0   68   190.0    94.0      29.0       105.0       40.8
1       1   20   166.0    60.0      14.0        94.0       40.3
2       0   69   179.0    79.0       5.0        88.0       38.7
3       1   34   179.0    71.0      13.0       100.0       40.5
4       1   27   154.0    58.0      10.0        81.0       39.8
```

```
Y.head()
```

```
0    231.0
1     66.0
2     26.0
3     71.0
4     35.0
Name: Calories, dtype: float64
```

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

Model Training

```
li=[i for i in range(30)]
models=[LinearRegression(),Lasso(),Ridge(),DecisionTreeRegressor(),XGBRegressor()]
accuracies=[]
max_acc=0
best_model=0
best_random_state=0
for k in models:
    model=k
    for i in li:
        model.fit(X_train,Y_train)
        y_pred=model.predict(X_test)
        acc=r2_score(Y_test,y_pred)
        if max_acc<=acc:
            max_acc=acc
            best_model=model
            best_random_state=i
        accuracies.append(round(acc*100,2))
print("Best Accuracy Acquired: ",max(accuracies))
print('Best Model: ',best_model)
print('Best Random State: ',best_random_state)
```

```
Best Accuracy Acquired:  99.88
Best Model:  XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=None, n_jobs=None,
             num_parallel_tree=None, random_state=None, ...)
Best Random State:  29
```

XGBoost Regressor

```
model = XGBRegressor()
```

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

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
```

Predictive Model

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

```
Gender = "Male"
Age = "20"
Height = "193.0"
Weight = "86.0"
Duration = "11.0"
Heart_Rate = "92.0"
Body_Temp = "40.4"
if Gender=="Male":
    Gender=0
else:
    Gender=1

input_list = np.array([int(Gender), float(Age), float(Height), float(Weight),
float(Duration), float(Heart_Rate), float(Body_Temp)])
input_list = input_list.reshape(1, -1)
calories = round(model.predict(input_list)[0],1)
print("Calories Spent Predicted:", calories)
```

```
Calories Spent Predicted: 37.7\
```

```
mae = metrics.mean_absolute_error(Y_test, test_data_prediction)
print("Mean Absolute Error = ", mae)
```

```
Mean Absolute Error =  1.4833678883314132
```