DA 213 Python Programming Laboratory
Project Report
FitAI : Your Fitness Partner

by

Saptarshi Mukherjee
Ishan Chandra Gupta
Soumya Savarn

1. Introduction

FitAI is an innovative Python lab project designed to combine the power of generative AI and classical machine learning algorithms to predict fitness-related outcomes. This report presents a predictive modeling approach using regression to determine the exercise regimen necessary for individuals to achieve their fitness goals. By leveraging user data such as age, weight, height, dietary habits, medical history and existing fitness levels, the model aims to provide tailored recommendations for optimal exercise routines. This comprehensive report aims to provide insights into various aspects of the project, including a literature review, mathematical foundations, methodology, comparative results, discussions, and performance metrics.

2. Literature Review

In recent years, there has been a notable increase in the development of fitness tracking applications and predictive analytics within the health and wellness sector. Various studies have underscored the significance of employing machine learning algorithms for personalized fitness predictions.

Websites we referred to:

1. https://www.thehealthybeat.org/caloriebudget.aspx
2. https://www.kaggle.com/datasets/aadhavvignesh/calories-burned-during-exercise-and-activities .

3. Mathematical Foundations

The FitAI project is grounded in mathematical principles drawn from both classical machine learning and genetic algorithms. Classical ML algorithms, such as regression, classification, and clustering, are fundamental to predictive analysis, while genetic algorithms are utilized to optimize fitness goals based on user preferences and historical data. Regression analysis forms the basis of the predictive model in this project and generative AI part comes to our rescue when we don't have enough data to train a model and also we want to incorporate the best features for our users like the automated calorie logging from the mess menu.

Some formulae used in Regression:

$$Y = Wx + e$$

where, Y is the target values,

W is the feature matrix(i.e. $W[i][j]$ is the weight associated to jth value of X vector),

e is the bias vector or error vector i.e. the difference between the predicted and the original values,

x is the weight vector representing the number of independent variables.

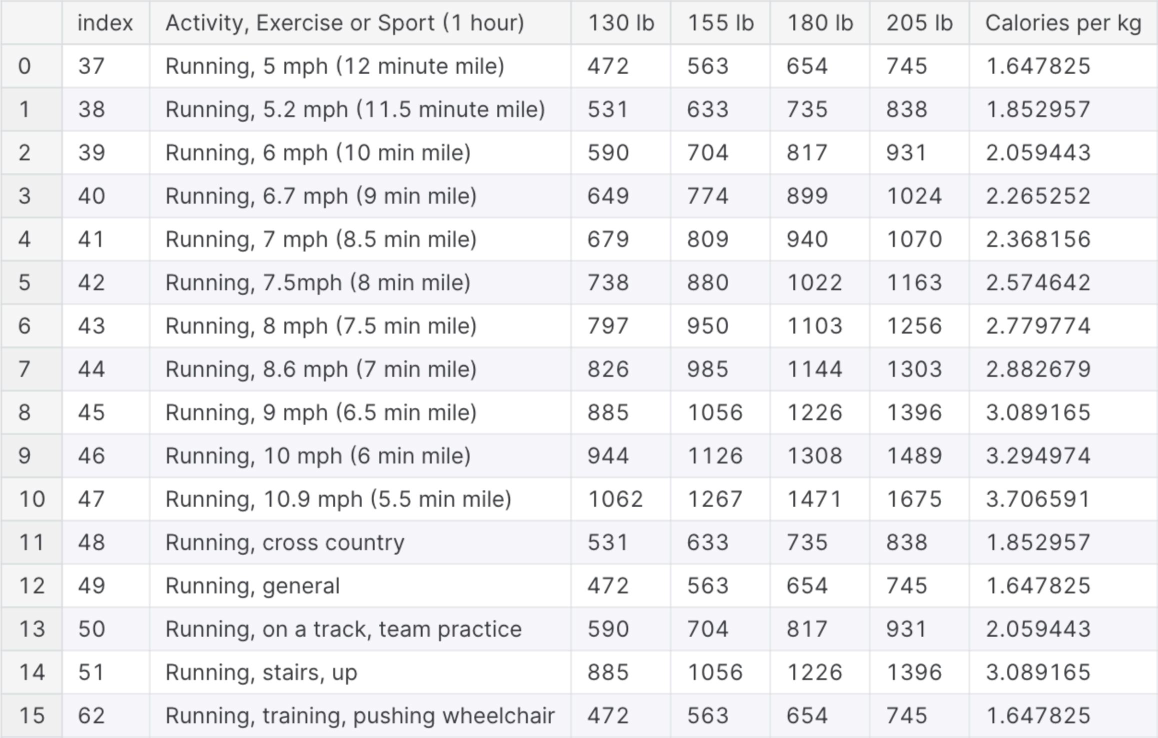
Our aim is to minimize the error ,i.e, minimize $||Wx - Y||$.

On solving, we get $x = inv\ (\ t(W)\ *\ W)\ *\ W\ *\ y$

Here, $inv(W)$ indicates the inverse of matrix W and $t\ (W)$ indicates the transpose of W.

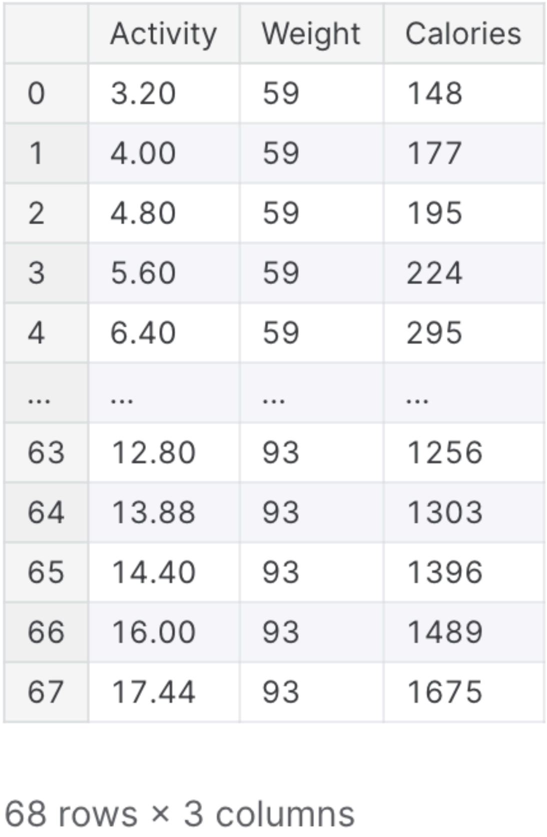
Description of the technique:

First, I isolated those rows in the dataset which give us the number of calories burnt every hour walking/running by people of different weights.



	index	Activity, Exercise or Sport (1 hour)	130 lb	155 lb	180 lb	205 lb	Calories per kg
0	37	Running, 5 mph (12 minute mile)	472	563	654	745	1.647825
1	38	Running, 5.2 mph (11.5 minute mile)	531	633	735	838	1.852957
2	39	Running, 6 mph (10 min mile)	590	704	817	931	2.059443
3	40	Running, 6.7 mph (9 min mile)	649	774	899	1024	2.265252
4	41	Running, 7 mph (8.5 min mile)	679	809	940	1070	2.368156
5	42	Running, 7.5mph (8 min mile)	738	880	1022	1163	2.574642
6	43	Running, 8 mph (7.5 min mile)	797	950	1103	1256	2.779774
7	44	Running, 8.6 mph (7 min mile)	826	985	1144	1303	2.882679
8	45	Running, 9 mph (6.5 min mile)	885	1056	1226	1396	3.089165
9	46	Running, 10 mph (6 min mile)	944	1126	1308	1489	3.294974
10	47	Running, 10.9 mph (5.5 min mile)	1062	1267	1471	1675	3.706591
11	48	Running, cross country	531	633	735	838	1.852957
12	49	Running, general	472	563	654	745	1.647825
13	50	Running, on a track, team practice	590	704	817	931	2.059443
14	51	Running, stairs, up	885	1056	1226	1396	3.089165
15	62	Running, training, pushing wheelchair	472	563	654	745	1.647825

Data after processing



	Activity	Weight	Calories
0	3.20	59	148
1	4.00	59	177
2	4.80	59	195
3	5.60	59	224
4	6.40	59	295
...	...	...	...
63	12.80	93	1256
64	13.88	93	1303
65	14.40	93	1396
66	16.00	93	1489
67	17.44	93	1675

68 rows × 3 columns

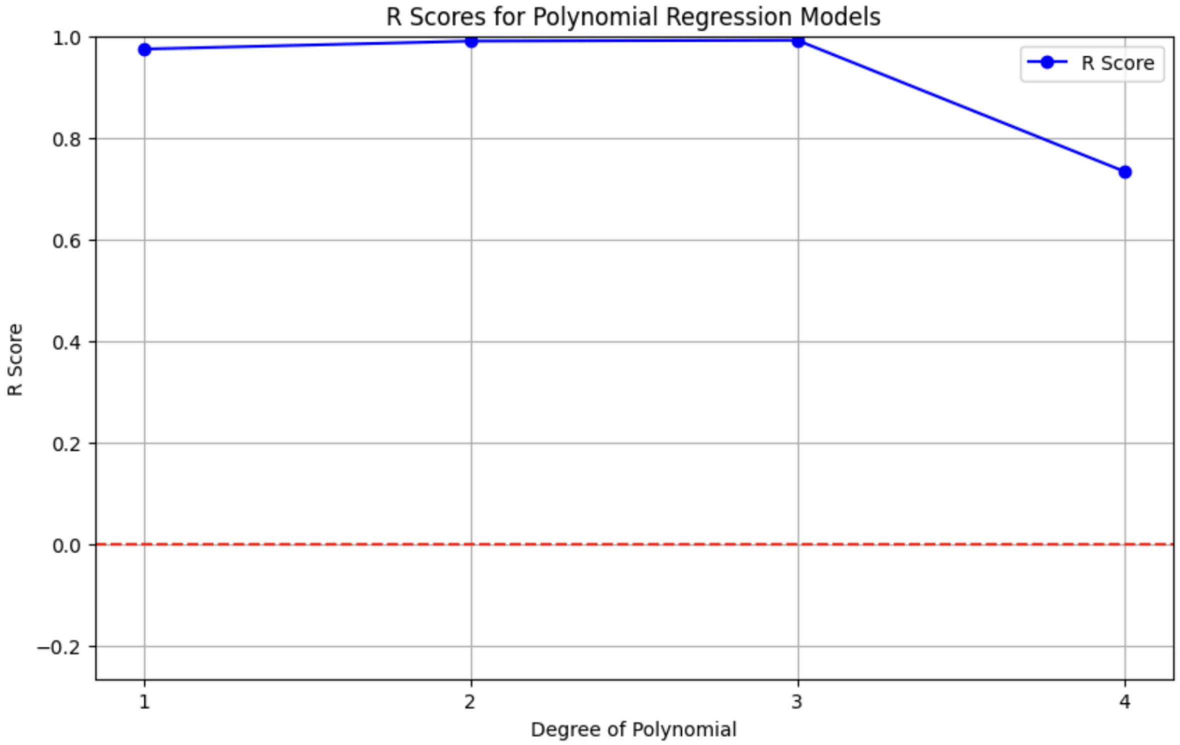
Data used in the ML model:
Features: Weight, Calories
To predict: Activity
Link to the notebook
https://www.kaggle.com/code/saptarshim596/python-lab-cp-re

Why higher degree regression didn't work well:

I tried higher degree regression but there just wasn't enough data to be able to fit a higher degree polynomial (which was in some cases, giving a lower training error, but there was a high test error so it was clear that there was overfitting occurring. In fact, regression with degree 2 or 3 was giving the best score(highest coefficient of determination-described later).

Output:

```
R Score for model with degree 1: 0.9744537020309676
R Score for model with degree 2: 0.990144071051682
R Score for model with degree 3: 0.991805723113886
R Score for model with degree 4: 0.7337080168658203
R Score for model with degree 5: 3.072505900875285
```

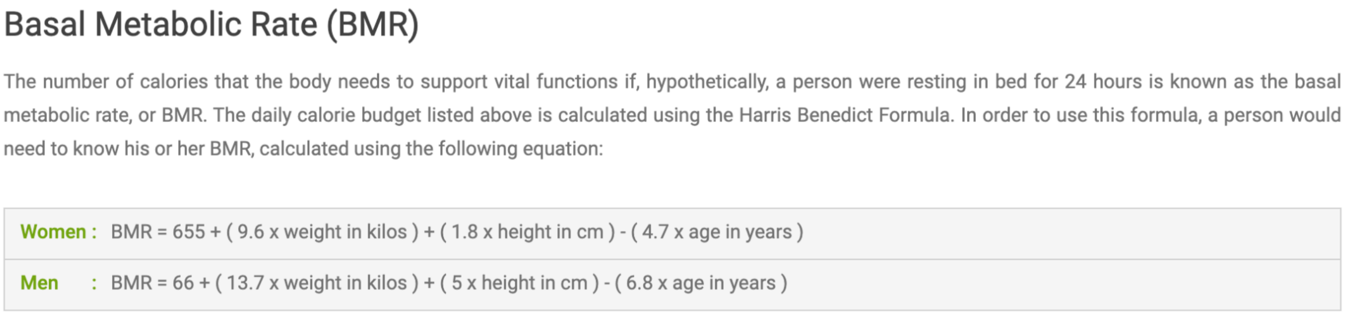


4. Methodology

Data Collection: User data including age, weight, height, dietary habits, exercise history, and fitness goals were collected through surveys and fitness tracking applications.

Data Preprocessing: The collected data underwent preprocessing steps such as normalization, missing value imputation, and feature scaling to ensure compatibility with the regression model.

The user's data is used to calculate his/her BMR(or basal metabolic rate) as per the following formulae:-



Basal Metabolic Rate (BMR)

The number of calories that the body needs to support vital functions if, hypothetically, a person were resting in bed for 24 hours is known as the basal metabolic rate, or BMR. The daily calorie budget listed above is calculated using the Harris Benedict Formula. In order to use this formula, a person would need to know his or her BMR, calculated using the following equation:

Women :	BMR = 655 + (9.6 x weight in kilos) + (1.8 x height in cm) - (4.7 x age in years)
Men :	BMR = 66 + (13.7 x weight in kilos) + (5 x height in cm) - (6.8 x age in years)

We used the user’s activity level to multiply a factor > 1 to the BMR, as the BMR is the number of calories needed to support vital functions for a person who is at bed rest.

4.1 Features Incorporated:

- **Activity Tracking:** FitAI captures and processes data user input to monitor daily activities.
- **User Calorie Intake:** FitAI uses google generative AI to process calorie intake of a user from an image(as an input provided by the user) and the user can also provide data manually.
- **Goal Setting:** Users can establish personalized fitness objectives tailored to their current activity levels and desired outcomes.
- **Predictive Analysis:** FitAI employs ML algorithms to forecast future fitness achievements and milestones. It uses a linear regression model to predict the distance and the speed at which the user needs to walk that distance each day to reduce his/her weight according to his/her wish.
- **User-Friendly Interface:** The application boasts an intuitive interface designed for seamless navigation and ease of use.

4.2 Integration of Tools and LIbraries:

- **CS50 Integration:** Concepts from Harvard's CS50 course are implemented to ensure the system's robustness and reliability. It provided foundational functions and utilities for web development using Flask.
- **Flask Framework:** Flask is a web application framework written in python. Flask is based on the WGSI toolkit and jinja2 template engine. FitAI is built using Flask which provides higher readability to the users.

- **Flask-Session:** Secure storage of session variables is implemented to maintain user state across requests.

- **Requests:** FitAI interacts with external APIs to fetch relevant data and enhance the user experience.

- **Cachelib:** Caching mechanisms are utilized to optimize performance and reduce latency.

- **Google Generative AI:** Cutting-edge AI technology is leveraged to generate personalized fitness recommendations.

- **Pillow:** Image processing tasks such as resizing and cropping are performed to enhance visual appeal.

- **Matplotlib:** Matplotlib is a comprehensive library for creating static,animated,and interactive visualizations in python.

Step by step utilization:

Step 1: The user registers and provides his basic information (like age, height, weight, target weight, gender, illness etc).

Step 2: The user needs to log the number of calories he/she is consuming. This information will be used while generating the fitness plan. The user can also log his/her weight and exercise (like distance covered- which is easily obtained from wearable watches running fitness apps)

Step 3: The user can generate a fitness plan as per his/her requirement- the user enters the number of days he/she wants the plan to run and the number of minutes he/she is willing to spend each day. The fitness plan automatically caters to the need of the user, if the user has a serious illness, the plan will cap the maximum speed the user needs to walk at. The distance the user needs to walk each day also has an upper limit, the plan will take this into account as well.

Step 4: User can view the fitness plan, and log his/her progress in the exercise log.

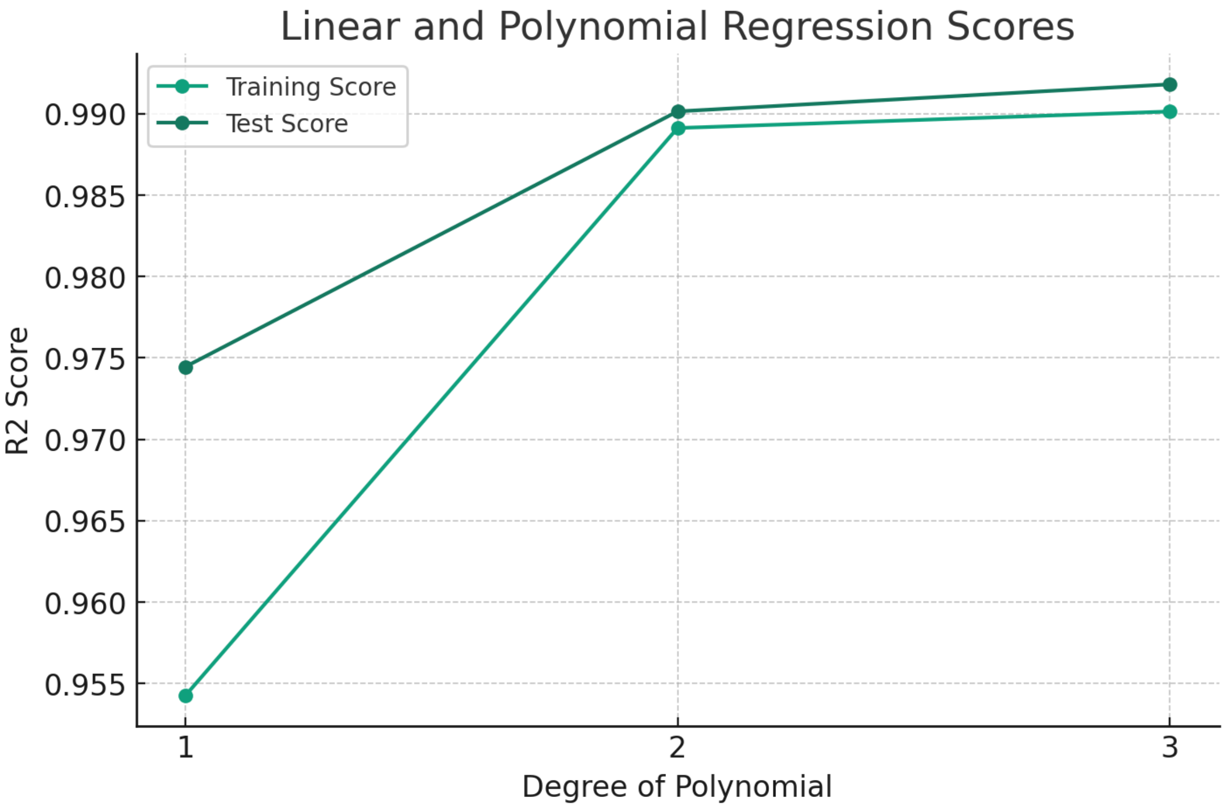
Step 5: The user can view his/her calorie consumption, exercise routine and also view the progress in weight loss with the help of graphs.

5. Comparative Results

Evaluation Metric used is the coefficient of determination(described later, under Performance metrics). Note that a score of 1 is best.

```
Training set linear regression score (for degree 1) is:
0.9542510422095909
Test set linear regression score (for degree 1) is:
0.9744537020309676
Training set polynomial regression score (for degree 2) is:
0.9891220402193815
Test set polynomial regression score (for degree 2) is:
0.990144071051682
Training set polynomial regression score (for degree 3 ) is:
0.9901354562247761
Test set polynomial regression score (for degree 3) is:
0.991805723113886
```

Notice that the **test set scores for models of degree 2 and 3 are very similar** (in fact the score for model with degree 3 is slightly better). However **we decided to use the model with degree 2 as the predictions obtained using that have a lower variance**. A model using a polynomial with **higher degree (3 here) might end up giving large outputs for extreme values**. This problem is far less for models with degree 2.



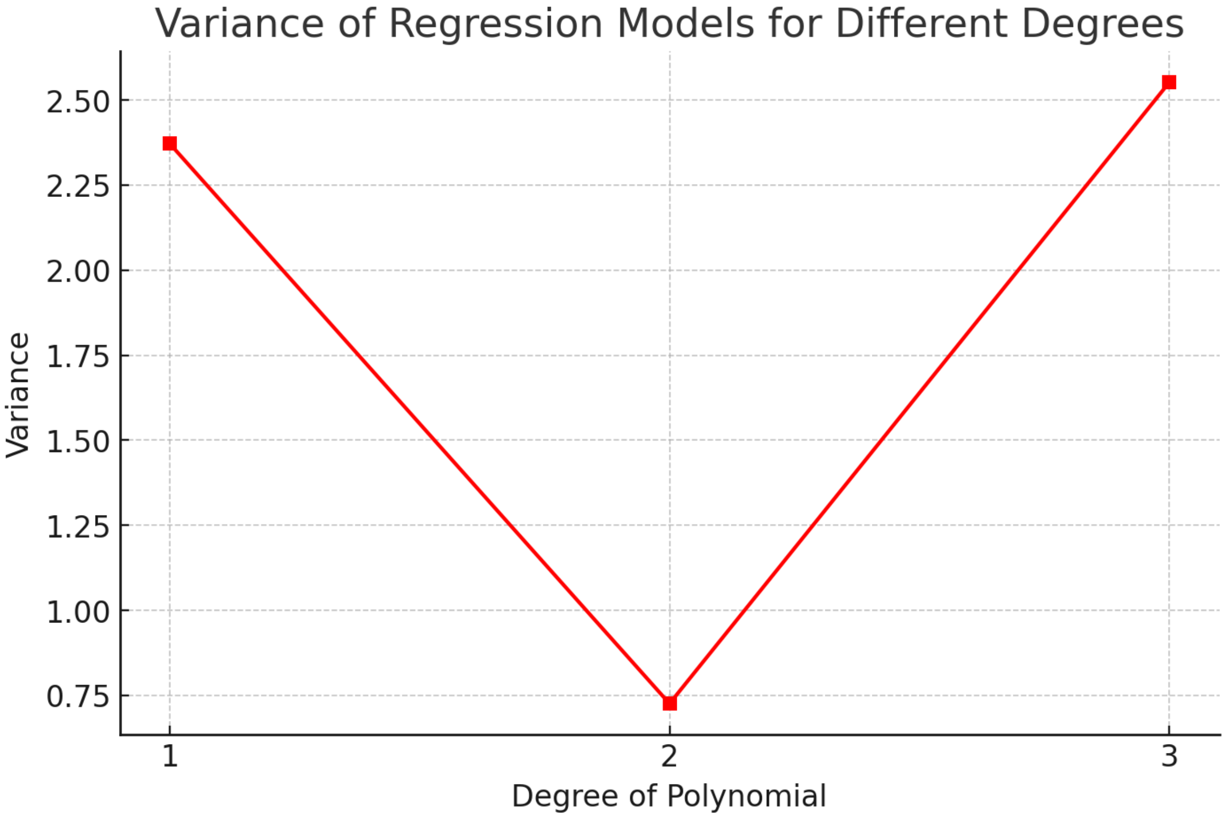
Now I tested 5 data points on 3 models, models with degree 1,2,3, and found that variance(in the predictions) is the lowest for the model with degree 2, which means that compared to the model with degree 3, it will give more stable values, which is a good thing.

Data:

```
val = np.array([[80, 340], [100, 450], [60, 250], [80, 410], [75,
290]])
```

Final output:

```
Variance for degree 1 is 2.3717292836926833
Variance for degree 2 is 0.7257377760320639
Variance for degree 3 is 2.551798466461758
```



6. Discussions

6.1. Challenges Faced:

There was not a lot of data available, so finding usable and relevant data was a challenge. Hyperparameter tuning(like setting the degree of the regression) was also an important task. For some datasets, it was possible to fit the dataset very well but then overfitting was a huge issue, as performance of a model trained on such data on new data was not very good.

6.2 How we incorporated the Gen-AI in our project:

Our project provides the user with an option to upload their diet chart and his/her calorie intake will automatically be calculated. This has been possible with the help of Google generative AI. We are creating a prompt with a message and an image input to get calories for a particular day.Then Gen-AI returns a response through which we extract our final answer as per the user input.

6.3 User Data Privacy and Security:

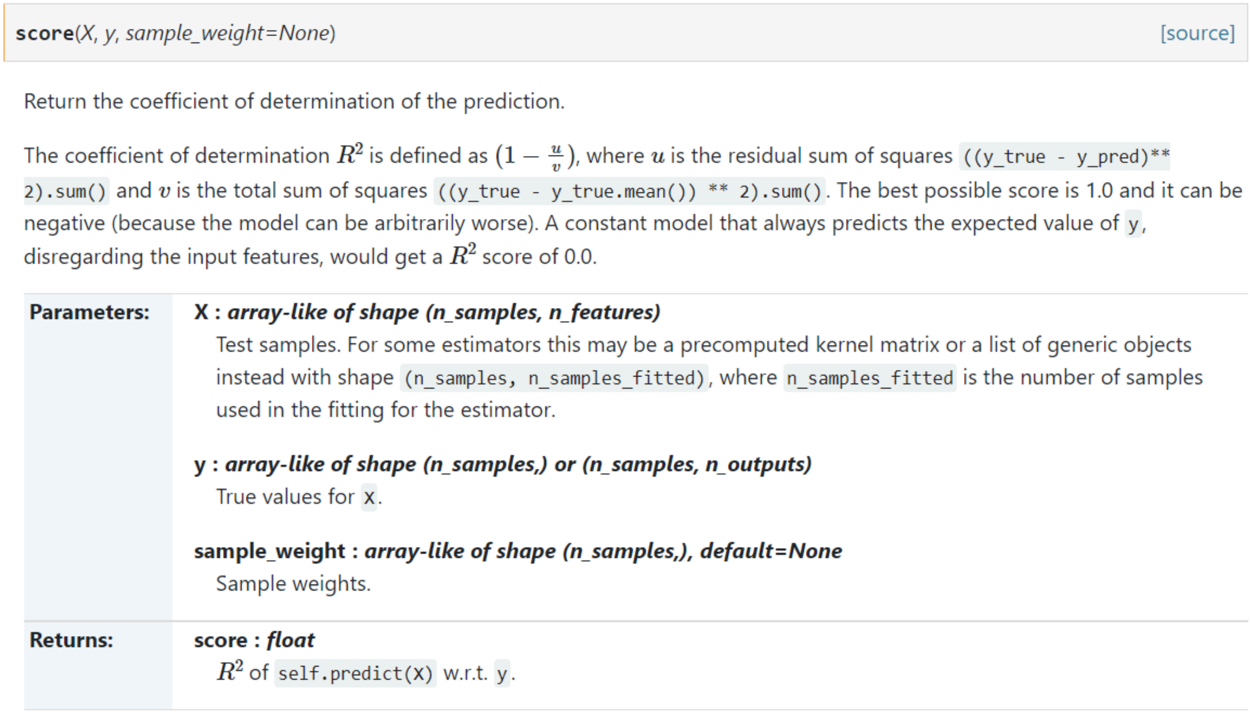
Minimize the collection of personal data is what we intend. We have also used encryption, secure protocols, and access controls to protect data from unauthorized access. We are storing hashed passwords of the users.Also we are transferring only essential information to Gen-AI for our purpose which is the mess-menu of the user. Users can access their own data only. Thus we focus on prioritizing user data privacy and security and also ensure compliance with legal and regulatory requirements.

6.4. Future Directions:

Future enhancements for FitAI include incorporating more advanced ML techniques, enhancing user interactivity and personalization, and exploring real-time data analysis capabilities for immediate feedback.

7. Performance Metrics:

The coefficient of determination (R score) commonly referred to as the R-squared (R^2) score was the performance metric we used. A score of 1.0 is the best possible score.



score(*X*, *y*, *sample_weight=None*) [source]

Return the coefficient of determination of the prediction.

The coefficient of determination R^2 is defined as $(1 - \frac{u}{v})$, where u is the residual sum of squares `((y_true - y_pred)** 2).sum()` and v is the total sum of squares `((y_true - y_true.mean()) ** 2).sum()`. The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of `y`, disregarding the input features, would get a R^2 score of 0.0.

Parameters:

X : *array-like of shape (n_samples, n_features)*
Test samples. For some estimators this may be a precomputed kernel matrix or a list of generic objects instead with shape `(n_samples, n_samples_fitted)`, where `n_samples_fitted` is the number of samples used in the fitting for the estimator.

y : *array-like of shape (n_samples,) or (n_samples, n_outputs)*
True values for `X`.

sample_weight : *array-like of shape (n_samples,), default=None*
Sample weights.

Returns:

score : *float*
R^2 of `self.predict(X)` w.r.t. `y`.

8. Conclusion

The FitAI project demonstrates the potential of integrating classical machine learning and genetic algorithms for personalized fitness tracking and prediction. Through a combination of robust methodologies, advanced technologies, and innovative features, FitAI aims to revolutionize the way individuals approach their fitness goals.

References:

1. CurFi: An automated tool to find the best regression analysis model using curve fitting
2. https://www.thehealthybeat.org/caloriebudget.aspx