Numerical Methods
24311

Calorie Expenditure Estimation Using Polynomial Regression

Keith Boyle, Dustin Moss, Parker Owens, Evan Roper, Mitchell Stokey

Meet the Team

Keith Boyle

Dustin Moss

Evan Roper

Mitchell Stokey

Parker Owens

Table of contents

01

Background on our Project

Description of our project and calorie expenditure

Motivation

Our motivation is to understand how our body expends energy during intervals of exercise

Using polynomial regression and gradient descent we trained a machine learning model to calculate the average amount of calories burned based on:

What is a Calorie?

A calorie is:

- Calories are a measure of energy.
- Amount of energy required to raise the temperature of exactly one gram of water by one degree Celsius at one atmospheric pressure
- Kilogram calories (Cal), are more commonly known and refer to the calories in food.

How are Calories Measured?

The Science

- amount of energy required to raise the temperature of exactly one gram of water by one degree Celsius at one atmospheric pressure

Simplified

- The amount of energy that we need to keep our bodies happy, healthy, and strong

02

Explanation of our Model

Analysis of our data, model, and methods

Multiple Variable Linear Regression

Multiple Variable Linear Regression is a machine learning method that uses a linear equation to model a mathematical relationship among several input variables to predict an output. The model learns optimal coefficients using an optimization algorithm to minimize a cost function.

Multi Linear Regression

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

Multiple Variable Polynomial Regression

Multiple Variable Polynomial Regression is a machine learning method that uses a polynomial equation instead of a linear equation to model relationships between multiple input variables and an output variable.

This allows for better predictions when feature data does not linearly reflect a specific output.

2nd Degree Multi Polynomial Regression

$$y = b_0 + b_1x_1 + b_2x_1^2 + \dots + b_{2n-1}x_n + b_{2n}x_n^2$$

Our Data

Our dataset was obtained from an online database called Kaggle.com. Our data specifically was obtained from a variety of athletes and nonathletes. Several men on the football team were recorded as well as students apart of various clubs.

https://www.kaggle.com/datasets/fmendes/fmendesdat263xdemos/discussion/476313

This data has 15,000 instances with data on age, height, weight, workout duration, average heart rate, average body temperature, sex, and calories burnt.

(However, this is an older data set that only accounted for men and women. There would have to be new data collected for anyone who identifies differently.)

Feature Engineering

Feature engineering is transforming raw data into informative and useful features to improve machine learning model performance.

1

One-Hot Encoding:

- Handles categorical features
- Makes usable in regression

Sex
Male
Female
Female

→

Male	Female
1	0
0	1
0	1

2

Z-Score Scaling (Standardization):

- Scaling each feature to a similar range
- Aids in gradient descent convergence
- Allows weights of each coefficient to be analyzed

03

Preliminary Results and Applications

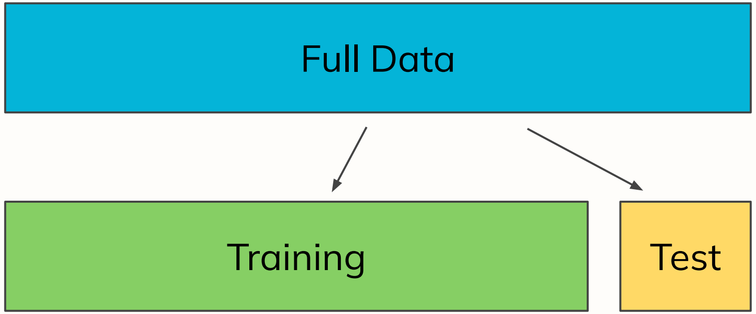
Exploring and explaining our results and applications

Quick Code Overview

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                              DATA MANIPULATION                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loading data
raw_data_table = readtable("full_data_set.csv");

% Converting "Gender" col to "Male" "Female" boolean cols
gender = raw_data_table.Gender;
raw_data_table.Male = strcmpi(gender, 'male');
raw_data_table.Female = strcmpi(gender, 'female');
raw_data_table.Male = double(raw_data_table.Male);
raw_data_table.Female = double(raw_data_table.Female);
raw_data_table(:, 'Gender') = []; % Remove the original "Gender" col

% Preparing Data for Training
raw_data = table2array(raw_data_table);
% raw_data = raw_data(randperm(size(raw_data, 1)), :); % Randomizing rows
data = [raw_data(:,2:7), raw_data(:,9:10)]; % Input variables
data = normalize(data); % Normalizing
cal = raw_data(:,8); % Output variable

% Add polynomial features
degree = 2; % Change the degree as needed
data_poly = add_polynomial_features(data, degree);
```

```
% Spliting training and test
split = 0.90; % Percent of training data
split_index = round(split.*size(data_poly,1));
train_data = data_poly(1:split_index,:);
train_cal = cal(1:split_index,:);
test_data = data_poly(split_index+1:end,:);
test_cal = cal(split_index+1:end,:);

% Training
theta = zeros(size(train_data,2), 1); % Parameters
alpha = 0.01; % Learning rate
num_iters = 10000; % Number of iterations
[theta, J_history] = gradientDescent(train_data, train_cal, theta, alpha, num_iters);

% Prediction
pred_cal = test_data * theta;

% Plot the cost function over iterations to see if it's decreasing
figure;
plot(1:num_iters, J_history, '-b', 'LineWidth', 2);
xlabel('Number of iterations');
ylabel('Cost J');
title('Cost function over iterations');

% Display the trained parameters
feature_names = raw_data_table.Properties.VariableNames([2:7, 9:10]);
% fprintf('Trained Parameters:\n\n'); ...
```

Splitting The Data




```
% Spliting training and test
split = 0.90; % Percent of training data
split_index = round(split.*size(data_poly,1));
train_data = data_poly(1:split_index,:);
train_cal = cal(1:split_index,:);
test_data = data_poly(split_index+1:end,:);
test_cal = cal(split_index+1:end,:);
```

The model's coefficient weights are learned with the training set that has info on calories burned

The model then uses the test set to predict calories burned and compare to actual
- evaluates model performance on unseen data

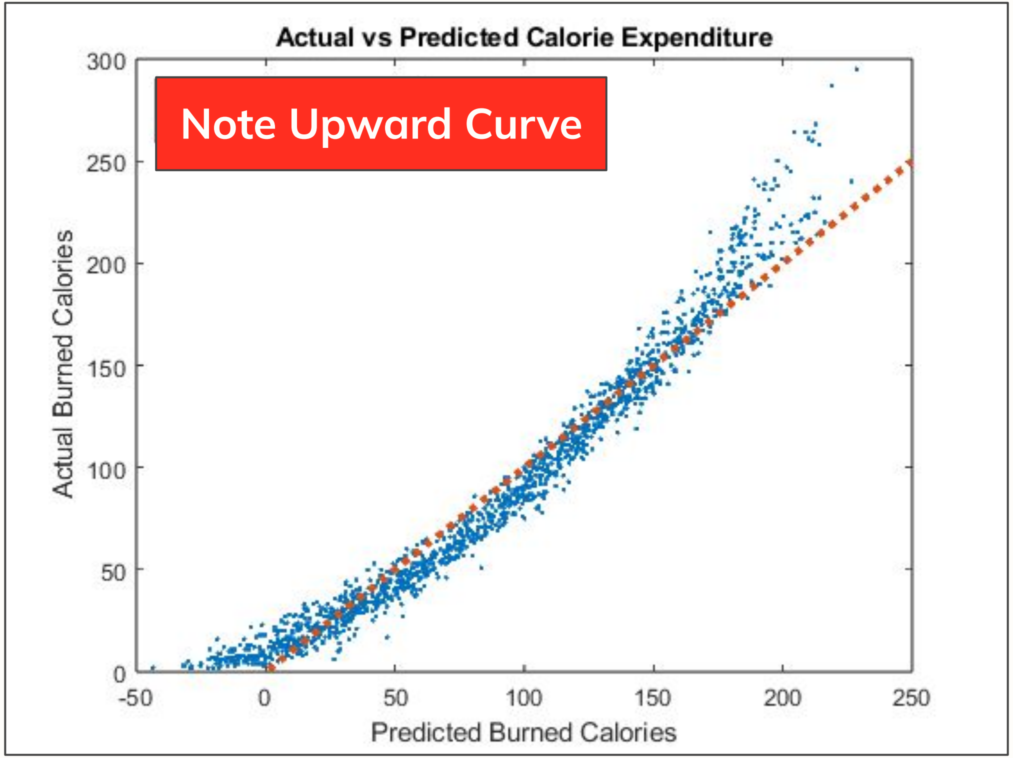
Quick Code Overview Cont.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                ERROR ANALYSIS                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure();
plot(pred_cal,test_cal,'.','LineWidth', 1);
xlabel('Predicted Burned Calories');
ylabel('Actual Burened Calories');
title('Actual vs Predicted Calorie Expenditure');
hold on
plot(1:250,1:250,":",'LineWidth',3)

% Calculate R-squared value
mean_cal = mean(test_cal);
SSTO = sum((test_cal - mean_cal).^2);
SSE = sum((pred_cal - test_cal).^2);
R_squared = 1 - (SSE / SSTO);
fprintf('R-squared value: %.5f\n', R_squared);
```

```
function [theta, J_history] = gradientDescent(X, y, theta, alpha, num_iters)
    m = length(y);
    J_history = zeros(num_iters, 1);
    for iter = 1:num_iters
        predictions = X * theta;
        errors = predictions - y;
        delta = (1/m) * (X' * errors);
        theta = theta - alpha * delta;
        J_history(iter) = computeCost(X, y, theta);
    end
end
```

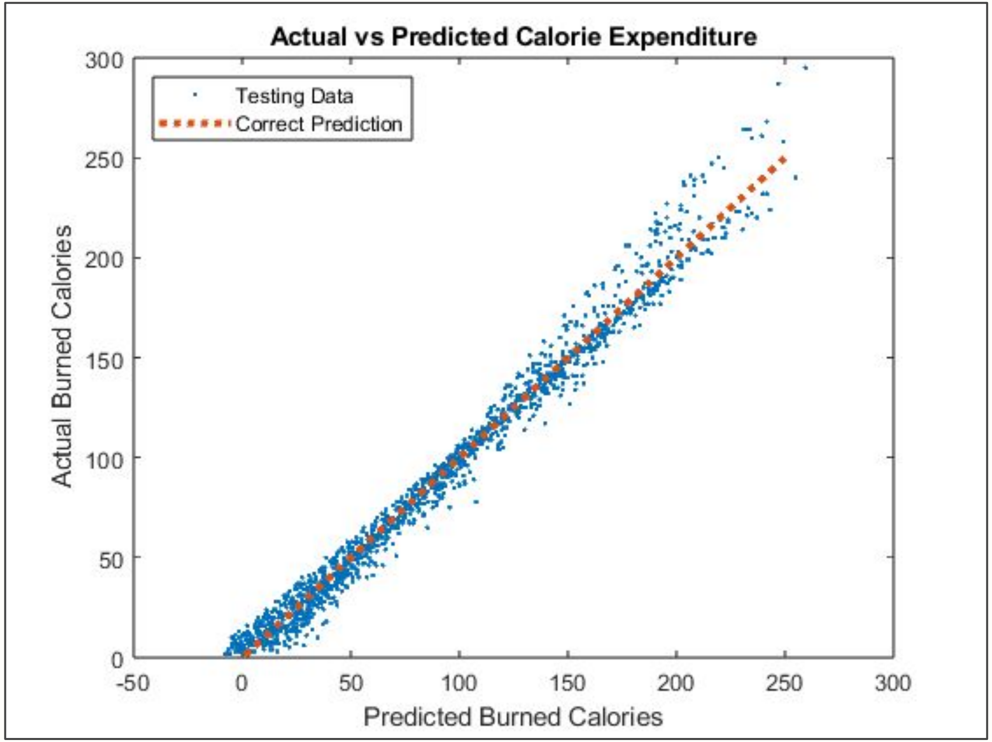
```
function J = computeCost(X, y, theta)
    m = length(y);
    predictions = X * theta;
    squared_errors = (predictions - y).^2;
    J = 1/(2*m) * sum(squared_errors);
end
```

Initial Model (1st Order)




```
Trained Parameters

 Regression Equation:

 + 89.54667
 + 8.45926(Age)
 - 2.59045(Height)
 + 4.48656(Weight)
 + 55.26259(Duration)
 + 19.06263(Heart_Rate)
 - 13.27912(Body_Temp)
 - 0.30773(Male)
 + 0.30773(Female)
 = Calories Burnt

R-squared value: 0.96524
```

Improved Model (2 Order)




```
Trained Parameters

 Regression Equation:

 + 25.60044
 + 8.64302(Age)
 - 0.33622(Height)
 + 0.78174(Weight)
 + 42.60766(Duration)
 + 19.26225(Heart_Rate)
 + 0.69425(Body_Temp)
 - 0.17447(Male)
 + 0.17447(Female)
 + 0.00639(Age^2)
 - 1.50801(Height^2)
 + 3.78822(Weight^2)
 + 5.11787(Duration^2)
 + 5.10728(Heart_Rate^2)
 + 0.24110(Body_Temp^2)
 + 25.59627(Male^2)
 + 25.59627(Female^2)
 = Calories Burnt

R-squared value: 0.98262
```

Optimal Model Parameters and Results

Order	Learning Rate	Max Iterations	R-Squared
First	0.01	10,000	95.687%
Second	0.005	20,00	98.262 %
Third	0.008	15,000	98.248 %
Fourth +	-	-	NaN

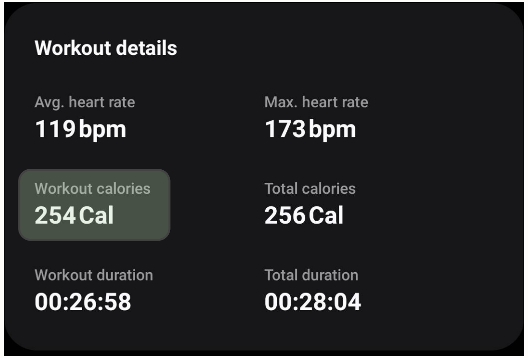
2nd Order Performed the Best

4th Order + Diverged
- More Tuning needed
  - learning rate
  - number of iterations
- Change opt. function

Testing on our Own Data!

- Asked various participants to record a workout on their respective device of choice and to report data on their workout and self
- Requested participants took body temp orally 3 times during workout and to record these values for us.
- Interested in "Active Calories" prediction

Biking with Samsung Watch



Hiking with a Fitbit

```
Enter your age: 20
Enter your height (in ft'in"): 5'0"
Enter your weight (lbs): 121
Enter the duration of activity (min): 13.63
Enter your average heart rate (bpm): 147
Enter your average body temperature (°F): 100.9
Enter your sex (male/female): female

Estimated calories burnt: 152.76
```

00:13:38 Workout duration	00:13:58 Total duration
1.01 mi Distance	
146 Cal Workout calories	147 Cal Total calories
4.4 mi/h Avg. speed	6.7 mi/h Max. speed
13'29" /mi Avg. pace	08'52" /mi Max. pace
147 bpm Avg. heart rate	172 bpm Max. heart rate

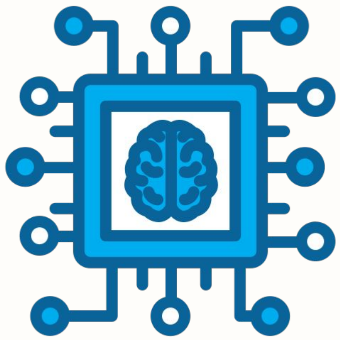
Lifting with Apple Watch

```
Enter your age: 21
Enter your height (in ft'in"): 6'5"
Enter your weight (lbs): 255
Enter the duration of activity (min): 42.58
Enter your average heart rate (bpm): 131
Enter your average body temperature (°F): 99.9
Enter your sex (male/female): male

Estimated calories burnt: 567.00
```

This caloric expenditure is significantly higher than most training data seen in the model but it was able to generalized pretty well to very intense unseen workouts!

Applications of our Model

- Gauge how hard you worked during a workout
- To guide athletes calorie intake
- Help with nutrition and dieting



04

Future Expansions and Work

Looking forward into expanding and improving this model

How can we Improve our Model

Cross Validation

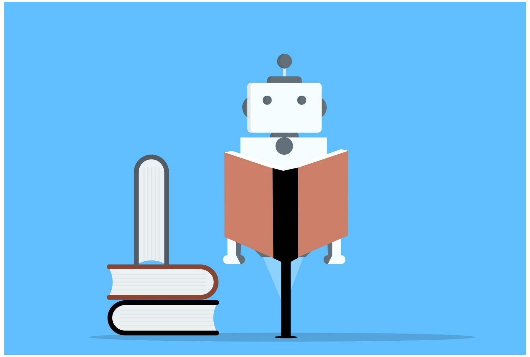
- Predict how well model works with other data sets
- K Folding

Regularization

- Adding a “Penalty” to coefficients
- Avoids overfitting
- Simpler Model

Other Data Sets

- See model’s performance on other data sets
- Experimental testing through smart watches



Future Considerations

Apps/Wearable Devices

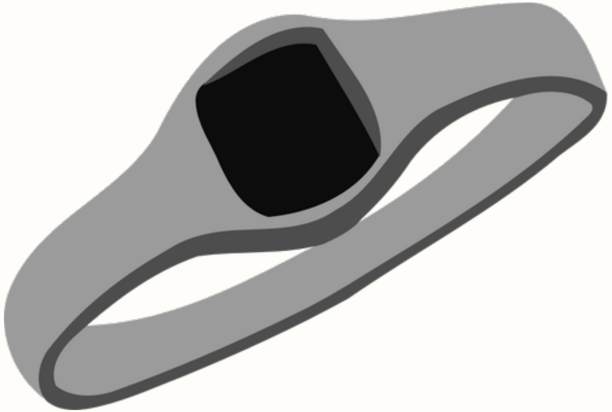
- Allow consumers to track workouts
- Benefits health and leads to smart life choices

Sports Performance

- Allows coaches to tailor workouts to specific player
- Increased focus on player safety and health

AI Integration

- Personalized workouts
- Reinforced Learning



Thanks!

Questions?