Indian Institute of Technology Indore

B.Tech., Mathematics & Computing

MA 212 – Regression Analysis

Calories Burnt Prediction

Predict-a-Burn

Project Report

Team Members

Anmol Jain	**230008009**
Nidarsana M	**230004031**
Salaj Bansal	**230002063**

April 2025

1 Overview

Many people keep track of their daily calorie intake and how much of it is used up in day-to-day activities. While directly monitoring the exact number of calories burnt can be challenging, predictive modeling offers a practical solution. By leveraging physiological and biometric data—such as age, weight, heart rate, and body temperature, collectively referred to as (**"features"**)—a regression model can be trained to estimate calorie expenditure (**"target"**) with reasonable accuracy.

This regression model is trained using datasets where the values of the features along with the actual burnt calories are tabulated. The actual burnt calories can be predicted using a few ways but the most common one in labs is using indirect calorimetry. In this process, oxygen consumption (VO_2) and carbon dioxide production (VCO_2) is measured using a metabolic mask using which, calories burnt can be obtained using the formula:

$$\text{Calories burnt} = (VO_2) \times \text{Caloric equivalent} \times \text{duration}$$

where,

$$\text{Caloric equivalent} \approx 5\,\text{kcal/L O}_2$$

Importance of Calorie Burn Prediction

Calorie burn prediction is essential for multiple fields, including:

- **Health and Fitness**: Helps individuals plan effective workouts and track their energy expenditure.
- **Weight Management**: Provides insights into energy balance, which is crucial for weight loss or maintenance.
- **Athletic Performance**: Enables athletes to optimize training intensity and recovery strategies.
- **Wearable Technology**: Enhances smart fitness devices by improving real-time calorie estimation.

Given the rising importance of personalized fitness, accurate models for calorie prediction can support AI-driven fitness coaches, meal planning, and goal-based exercise routines.

2 Project Objectives

The main objectives of this project are:

- **Develop a Predictive Model**: Build a model using a dataset that includes individual features such as age, height, weight, exercise duration, heart rate, and body temperature.

- **Identify Influential Factors**: Identify the key factors that contribute to calorie expenditure during physical activity.
- **Deliver a Machine Learning Solution**: Create a machine learning solution that accurately predicts the number of calories burned for unseen test data.

3 Techniques and Tools

- **Programming Language**: Python
- **Libraries Used**: Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn
- **Machine Learning Techniques**: Supervised Learning, Regression Models

4 Dataset Overview

Source: `https://www.kaggle.com/datasets/ruchikakumbhar/calories-burnt-prediction`
The dataset consists of 15,000 records, each representing an individual's workout session. It contains both physiological attributes and exercise-related parameters, making it ideal for calorie burn analysis. The primary columns include:

- **Calories**: The target variable representing the number of calories burned.
- **Gender**: Gender of the individual (Male or Female).
- **Age**: Age of the individual (in years).
- **Height**: Height of the individual (in cm).
- **Weight**: Weight of the individual (in kg).
- **Duration**: Duration of the exercise (in minutes).
- **Heart Rate**: Average heart rate during exercise (in beats per minute).
- **Body Temp**: Body temperature during exercise (in Celsius).
- **User_ID**: A unique identifier for each individual (dropped during preprocessing).

5 Analytical Framework

5.1 Data Exploration

Investigate the relationships between variables and identify any anomalies or patterns within the data. These include:

- Check for missing values
- Check for duplicate values

- Check the data type
- Check the statistics of the data

Analyzed the distribution of features such as age, weight, and calories burned using visualizations like histograms, scatter plots, and correlation heatmaps.

Key Analysis in Histogram

Feature-wise Inferences

- **Age**: Right-skewed; most users are in their 20s with density dropping as age increases.
- **Height**: Nearly normal; most fall between 160–190 cm.
- **Weight**: Slightly bimodal and symmetric; peaks around 60–80 kg.
- **Duration**: Roughly uniform with spikes at regular intervals, suggesting structured sessions.
- **Heart Rate**: Near-normal; majority between 85–110 bpm, typical for physical activity.
- **Body Temperature**: Slight right skew with a peak at 40–40.5°C, likely post-workout levels.
- **Calories**: Strong right skew; most burn less than 150 calories, few exceed 200.

Key Analysis in the Correlation Heatmap

- **Strong Positive Correlations (0.75–1):**
  - **Duration vs Calories Burned (0.96)**: Longer workout durations lead to significantly higher calorie expenditure.
  - **Heart Rate vs Calories Burned (0.99)**: A near-perfect correlation shows that intense sessions with higher heart rates burn more calories.
  - **Duration vs Heart Rate (0.85)**: Longer durations naturally elevate heart rate due to sustained effort.
  - **Duration vs Body Temperature (0.9)**: More extended physical activity increases body temperature due to energy expenditure.
  - **Height vs Weight (0.96)**: Taller individuals tend to weigh more — a typical anthropometric correlation.
  - **Body Temperature vs Calories Burned (0.82)**: Increased body heat aligns with greater energy (calorie) output.
  - **Heart Rate vs Body Temperature (0.77)**: Elevated heart rate corresponds with a rise in body temperature during exertion.

- **Moderate Positive Correlation (0.3–0.75):**
  - **Weight vs Calories Burned (0.35)**: Heavier individuals tend to burn slightly more calories, likely due to higher energy needs.
- **Weak Positive Correlations (0–0.3):**
  - **Age vs Calories Burned (0.15)**: Age has a small effect on calorie output, potentially due to metabolism changes.
  - **Height vs Calories Burned (0.018)**: Negligible influence of height on caloric burn.
  - **Weight vs Heart Rate (0.043)**: Slight tendency for heavier individuals to have marginally elevated heart rates.
  - **Weight vs Body Temperature (0.041)**: Minimal impact of weight on temperature regulation during activity.
- **Negative Correlations:**
  - **Age vs Height (-0.0096)**: Insignificant inverse trend, possibly linked to height loss with aging.
  - **Age vs Weight (-0.09)**: Slight weight decrease with increasing age.
  - **Height vs Duration (-0.046)**: Taller individuals might prefer shorter workout durations.
  - **Weight vs Duration (-0.019)**: A marginal trend suggesting heavier individuals may engage in shorter sessions.
- **Correlations Near Zero:**
  - **Age vs Heart Rate (0.01)**: No significant connection between age and heart rate.
  - **Height vs Body Temperature (0.012)**: Height has virtually no effect on body temperature during exercise.
  - **Weight vs Calories Burned (0.035)**: Practically no direct relation between weight and calories burnt.

5.2 Data Preprocessing

- **Encoding**: Convert categorical variables like gender into numerical format. Improve the performance of the data analysis model by representing categorical values as numbers. Here, we assign 0 to male and 1 to female.
- **Scaling**: Apply Min-Max Scaling to numerical features (e.g., height, weight, heart rate) to normalize the data and improve model performance.
- **Feature Selection**: Remove irrelevant columns (e.g., User_ID) and focus on the key features directly related to calorie burn.

Calculation of MI Score

Mutual Information (MI) score measures how much information one variable gives about another. It is like the coefficient of correlation but while the latter measures the linear relation between two variables, the former measures any kind of dependency between two variables. These important values are tabulated below:

S No.	Feature	Correlation Coefficient (with target)	Mutual Information Score
1	Age	0.15	0.025639
2	Height	0.017	0.009392
3	Weight	0.035	0.017669
4	Duration	0.96	1.515941
5	Heart Rate	0.90	0.877684
6	Body Temperature	0.82	0.990395
7	Gender	NA	0.008655

These results show that duration, heart rate, and body temperature are the most important and dominant features to predict the calories burnt.

5.3 Model Building

Splitting the Data

We will split the data into two parts:

- **X**: The features (all columns except for the target).
- **y**: The target column.

We divided the data into two main sets:

- **Training set**: Used to train the model and build it, where the data is used to learn patterns and relationships.
- **Testing set**: Used to evaluate the model's performance after training, where this data is used to assess the model's accuracy and effectiveness in making predictions.

Analysis of Regression Models

To determine the most effective regression model for calorie prediction, we evaluated several algorithms based on performance metrics and theoretical robustness. Below is a compact summary of the models considered:

- **Linear Regression**:

$$y = X\beta + \epsilon, \quad \beta = (X^T X)^{-1} X^T y$$

Assumes linearity, independence, homoscedasticity, normality, no multicollinearity, and exogeneity.

- **Lasso Regression**:

  $$J(\beta) = \|y - X\beta\|_2^2 + \alpha \sum |\beta_j|$$

  Encourages sparsity via L1 regularization.

- **Ridge Regression**:

  $$J(\beta) = \|y - X\beta\|_2^2 + \alpha \sum \beta_j^2, \quad \beta = (X^T X + \alpha I)^{-1} X^T y$$

  Penalizes large coefficients, handles multicollinearity.

- **K-Nearest Neighbors (KNN)**:

  $$\hat{y} = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$

  Based on local similarity using distance metrics.

- **Decision Tree**:

  $$\hat{y}(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

  Uses feature-based splits to minimize MSE.

- **Random Forest**:

  $$\hat{y}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$$

  Ensemble of trees trained on bootstrap samples and random features.

- **XGBoost**:

  $$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad J = \sum (y_i - \hat{y}_i)^2 + \sum \Omega(f_k)$$

  Gradient boosting with regularization, where $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|_2^2$.

- **CatBoost**:

  $$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

  Gradient boosting with ordered boosting and native categorical encoding.

- **AdaBoost**:

  $$\hat{y}(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

  Sequentially reweights weak learners to minimize errors.

5.3.1 Multiple Linear Regression (MLR)

Baseline Exploration

As the course emphasizes Multiple Linear Regression, we evaluated its potential for predicting calorie expenditure. While not the most advanced model, MLR serves as a baseline to assess linear relationships between features and the target.

Key results:

- $R^2 = 0.967$: Explains 96.7% of variance in the data.
- **Duration** ($\beta = 6.64, p < 0.001$) and **Heart Rate** ($\beta = 1.99, p < 0.001$) are the most significant predictors.
- **Weight** and **Age** have smaller but significant effects.
- **Height** ($\beta = -0.156$) has a slight negative impact on calories burnt.
- Condition number $= 2.66 \times 10^4$: Indicates potential multicollinearity.

Although MLR performs decently, we attempted to improve it via feature engineering before exploring more advanced models.

Feature Engineering to Enhance MLR

Three new features were introduced to capture deeper relationships:

- **BMI** $= \frac{\text{Weight}}{\text{Height}^2}$: Represents body composition.
- **Duration** × **Heart Rate**: Captures workout intensity over time.
- **Weight** × **Age**: Reflects interaction between metabolic rate and body weight.

Enhanced model results:

- $R^2 = 0.987$: Improved from the baseline.
- AIC/BIC values decreased, indicating better model fit.
- **Duration** × **Heart Rate** ($\beta = 0.1306, p < 0.001$) is now the most impactful feature.
- Condition number $= 7.27 \times 10^5$: Suggests increased multicollinearity.

Despite the slight performance boost, multicollinearity and limited generalization led us to transition toward hyperparameter tuning of advanced models such as CatBoost, XGBoost, KNN, and Voting Regressor.

5.3.2 Hyperparameter Tuning and Ensemble Results

To boost model performance, we tuned CatBoost, K-Nearest Neighbors (KNN), and XGBoost using `GridSearchCV` and `RandomizedSearchCV`. We also constructed a weighted ensemble using `VotingRegressor`.

1. CatBoost Regressor CatBoost was tuned over depth, learning rate, and iterations. The best configuration:

- `depth`: 8, `learning_rate`: 0.03, `iterations`: 500

It achieved the highest performance with a cross-validation R^2 of 0.99958.

Metric	Training Set	Test Set
RMSE	0.9159	1.1151
MAE	0.6946	0.7794
R^2 Score	0.9998	0.9997

2. K-Nearest Neighbors (KNN) KNN was optimized over $k = 2$ to 30, with best result at $k = 9$. Despite high R^2, error metrics were significantly worse compared to other models.

Metric	Training Set	Test Set
RMSE	4.4717	4.8757
MAE	3.2230	3.5884
R^2 Score	0.9948	0.9941

3. XGBoost Regressor XGBoost was tuned across a wide hyperparameter grid. The best configuration:

- `n_estimators`: 600, `max_depth`: 3, `learning_rate`: 0.2
- `min_child_weight`: 7, `gamma`: 0.4, `colsample_bytree`: 0.5

Metric	Training Set	Test Set
RMSE	1.2584	1.4654
MAE	0.9519	1.0885
R^2 Score	0.9996	0.9995

4. Ensemble: Voting Regressor To leverage the strengths of the top models, we built a weighted ensemble with the following configuration:

- `VotingRegressor` with weights: CatBoost (3), KNN (1), XGBoost (2)

Metric	Training Set	Test Set
RMSE	1.0896	1.4195
MAE	0.8218	1.0099
R^2 Score	0.9997	0.9995

Conclusion

After evaluating multiple regression models and applying hyperparameter tuning, the baseline CatBoost Regressor emerged as the most effective model. It achieved exceptional performance with an R^2 score of 0.9999 on both training and test sets, along with the lowest RMSE and MAE values. These results demonstrate CatBoost's ability to generalize well without overfitting, making it the most appropriate choice for accurate calorie expenditure prediction.

5.4 Model Interpretation and Analysis of CatBoost

Original Dataset (Sample)

Gender	Duration	Heart Rate	Body Temperature	Calories
Male	29	105	40.8	231
Female	14	94	40.3	66
Male	5	88	38.7	26
Female	13	100	40.5	71
Female	10	81	39.8	35

Step 1: Preprocessing – Shuffling and Bucketizing

We shuffle the dataset and split the target **Calories** into 2 buckets:

- Bucket 0: 3 least values
- Bucket 1: First 2 greater values

Gender	Duration	Heart Rate	Body Temp.	Calories	Rating Bucket
Female	13	100	40.5	71	1
Male	29	105	40.8	231	1
Male	5	88	38.7	26	0
Female	10	81	39.8	35	0
Female	14	94	40.3	66	0

Step 2: Ordered Target Encoding

Categorical variables (like Gender) are encoded based on prior label stats using:

$$\text{Encoding} = \frac{\text{curCount} + \text{prior}}{\text{maxCount} + 1}$$

Gender	Duration	Heart Rate	Body Temp	Calories	Rating Bucket	Encoded Gender
Female	13	100	40.5	71	1	0.05
Male	29	105	40.8	231	1	0.05
Male	5	88	38.7	26	0	0.525
Female	10	81	39.8	35	0	0.525
Female	14	94	40.3	66	0	0.35

Step 3: Initial Prediction and Residual Calculation

We begin with predictions initialized to 0. Residuals = Actual - Prediction.

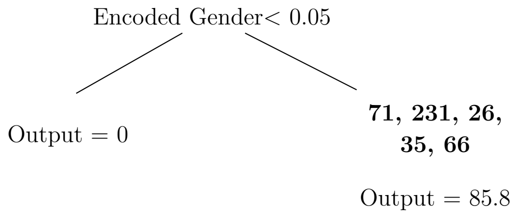
Encoded Gender	Duration	Heart Rate	Body Temp	Calories	Prediction	Residual
0.05	13	100	40.5	71	0	71
0.05	29	105	40.8	231	0	231
0.525	5	88	38.7	26	0	26
0.525	10	81	39.8	35	0	35
0.35	14	94	40.3	66	0	66

Step 4: Tree Split Using Encoded Gender

CatBoost tries different split points on encoded gender (e.g., 0.05, 0.35, 0.525) and evaluates their quality using cosine similarity between residuals and predicted outputs.

$$\text{Cosine similarity} = \frac{\sum A_i B_i}{\sqrt{\sum A_i^2} \cdot \sqrt{\sum B_i^2}}$$

where A and B are the two columns compared (Residual and Leaf output).
Example tree (Split: Encoded Gender < 0.05):



Step 5: Prediction Update

Predictions are updated using:

$$\text{New Prediction} = \text{Old Prediction} + \text{Learning rate}(\eta) \times \text{Leaf Output}$$

This step is repeated with new trees for other features (Duration, Heart Rate, etc.).

CatBoost Internals: Ordered Boosting

To avoid overfitting and target leakage:

- CatBoost uses **Ordered Boosting**, training each tree only on prior samples from a random permutation.
- This ensures gradient estimates are unbiased.

Why CatBoost?

- Handles categorical variables natively.
- Uses ordered target encoding and boosting to avoid overfitting.
- Delivers accurate results with minimal manual tuning.

This example walkthrough demonstrates how CatBoost constructs trees using smart encoding and unbiased boosting, delivering strong predictions with minimal preprocessing.

6 Result- Model Deployment and User Interface

Following model training and optimization, the CatBoost model was deployed in a web application named **Predict-a-Burn**. The interface takes user inputs — Height, Weight, Age, Duration, Heart Rate, Body Temperature, and Gender — and predicts calories burnt in kilocalories (kcal) using the trained model.

The app features a clean, user-friendly UI and provides instant results. This deployment demonstrates the practical value of our optimized model in a real-world setting.

The project is open-source and available at: `github.com/Anmoljain2005/Predict-a-Burn`.

7 Challenges

- **Feature Correlation**: High correlation between features like duration, heart rate, and calories may lead to multicollinearity.
- **Scaling**: Large value ranges (e.g., heart rate) can skew model performance without proper normalization.
- **Generalization**: Limited dataset size may affect model accuracy on unseen data.

8 Future Work and Enhancements

While the model performs well, future improvements could include:

- Including dietary intake to factor in energy balance.
- Using real-time wearable data to refine calorie estimates.

References

[1] Ruchika Kumbhar, *Calories Burnt Prediction Dataset*, Kaggle. Available at: `https://www.kaggle.com/datasets/ruchikakumbhar/calories-burnt-prediction`

[2] Prokhorenkova, L., et al., *CatBoost: Unbiased Boosting with Categorical Features*. Available at: `https://arxiv.org/abs/1706.09516`

[3] Pedregosa, F., et al., *Scikit-learn: Machine Learning in Python*. Available at: `https://scikit-learn.org/stable/about.html`

[4] James, G., et al., *An Introduction to Statistical Learning*. Available at: `https://www.statlearning.com/`

[5] Zhi-Hua Zhou, *Ensemble Methods: Foundations and Algorithms*. Available at: `https://scholar.google.com/citations?user=mkK-t00AAAAJ&hl=en`