NTNU

Kunnskap for en bedre verden

TDT4259 - Applied Data Science

Calorie Estimation for Cyclists

Peiretti Riccardo
Student ID: 133373

October, 2024

Table of Contents

1 One Pager

This document outlines a data science project that provides a potential solution for a hypothetical data-driven problem.

1.1 Intent

The document is designed to provide calorie estimation for cyclists, giving nutritional recommendations for training fueling. By analyzing various factors, the model estimates how many calories a cyclist needs to consume during training to optimize performance. It aims to deliver insights that help cyclists ensure they are properly fueled for every ride.

1.2 Desired outcome

The desired outcome is a Random Forest model which predicts a cyclist's ride caloric needs based on multiple variables. A realistic goal would be to:

- Avoid underfueling which can lead to fatigue and poor performance;
- Avoid overfueling which can reduce the performance;
- Achieve optimal energy levels for training.

1.3 Deliverable

The primary deliverable is a calorie estimation based on a Random Forest model that predicts the number of calories a cyclist needs to consume during a ride using a fake unreal dataset described in Section 6.2. This dataset incorporates input variables such as weight, height, fitness level, historical rides, training plans, while the output will be the recommendation for total calorie intake based on the inputs data.

1.4 Constraints

There are several constraints to take into consideration to reach the goal:

- **Athlete variability**: the developed model needs refining to reliably predict the response of individual patients to treatment [1];
- **Customer privacy**: no sensitive information of the athlete should be put at risk;
- **Complexity**: a Random Forest model creates a lot of trees which can make the model more complex and computationally expensive than a single decision tree, with longer prediction time compared to other models [2];
- **Environmental factors**: there are also external variables related to weather conditions that can influence the performance and they are not considered in the prediction.

2 Overview

This document presents a solution designed to help cyclists ensure they are properly fueled for every ride. It should be considered as an addition to existing fitness tracking platforms, complementing other systems that already monitor performance data. To achieve this goal, Random Forest should be a valid option due to its implementation typically in the field of healthcare and, furthermore, it plays a key role in the development of personalized recommendation systems [2]. This document tries to be a fundamental step in the application of new technologies in sports analytics.

3 Motivation

Nowadays, while tracking calorie intake is relatively easy, monitoring calorie burn is challenging due to the limited devices available [3]. In general, in the realm of endurance sports, athletes often grapple with the challenge of balancing caloric intake to avoid the dreaded energy depletion phenomenon known as '*bonking*' [4].

4 Success metrics

The success of the calorie estimation model is evaluated based on several key metrics, ensuring the model is accurate:

- **Prediction accuracy**: measure the accuracy of the Random Forest model by comparing predicted calorie requirements with actual energy consumption during rides;
- **Model generalization**: it is important generalize the model across different types of cyclists (e.g., beginner, intermediate, amateur, and professional), ensuring it provides accurate predictions regardless of fitness levels, body composition, or ride intensity.
- **User satisfaction**: collect feedback from athletes to understand how well the calorie predictions align with their experiences, particularly in terms of avoiding fatigue during rides. For example, positive feedback indicating fewer instances of '*bonking*' or overfueling will be considered a sign of success.

5 Requirements & Constraints

It is necessary to define, on the one hand, the *functional requirements* which are those that should be met to ship the project. On the other hand, *non-functional/technical requirements* which are those that define system quality and how the system should be implemented. Constraints can come in the form of non-functional requirements.

5.1 Functional requirements

The functional requirements focus on the primary goal of delivering accurate and personalized calorie recommendations. In the following Table 1, there are reported the functional requirements.

Requirement	**Description**
Calorie prediction	The model should accurately predict the number of calories a cyclist needs to consume during training, based on a specific training plan and corresponding training wattage zones.
User input	Cyclists will provide input data including their body metrics, planned ride details, training plan (e.g., duration, intensity), and historical ride data.
Recommendations	The output should be presented in a user-friendly format, providing cyclists with specific caloric intake recommendations to ensure they are properly fueled during the training session.

Table 1: Functional Requirements

5.2 Constraints

The constraints define the quality of the model and how it should perform. Table 2 below outlines these constraints.

Constraints	Description
Performance	Throughput: 1.000 predictions per second at peak load;
	Latency: 500 ms per request to ensure fast response times for athlete;
	Accuracy: 85% accuracy, with a maximum absolute prediction error equal to 300 kcal.
Cost efficiency	The cloud infrastructure cost for running the model should not be more than 100.000 NOK per month.
Data privacy	GDPR Compliance: the system must fully comply with GDPR and other relevant data privacy regulations, particularly ensuring user consent for data collection and processing;
	Anonymization: all Personally Identifiable Information (PII) must be anonymized or pseudonymized before being used in the model, ensuring that user identities cannot be linked to the data used for predictions.

Table 2: Constraints

5.3 What's in-scope & out-of-scope?

Given the complexity of the problem, it is important to define the boundaries of the project. This section defines what is included in the model and what is left for future developments. In fact, the model can deliver value to athletes without solving too many problems at once. Further details can be found in Section 6.

In-scope

- Delivering calorie estimates as the primary output, using body metrics, historical performance data, and training plans;
- Data collection, cleaning, and processing, along with feature identification;
- Model development;
- Experimentation and validation of the model.

Out-of-scope

- Post-training calorie recommendations or recovery strategies;
- Real-time in-ride fueling adjustments based on live feedback (e.g. as conditions change during the ride);
- A meal planning with macronutrient tracking from the calorie estimates.

First, the model does not take into consideration the athlete's feedback on the accuracy of the prediction: it could be integrated into future model which will be more complex, improving personalizing and making user-specific adjustments based on recurring patterns. Second, the current model focuses only on predicting calorie intake during the ride without recommending specific types of food (e.g., energy bars, gels, maltodextrins, sports salts). So, it does not include macronutrinet division or real-time fueling adjustments based on live feedback.

5.4 Assumptions

For this project, it is assumed that a fake unreal dataset with sufficient and high-quality historical data is available for model training and validation, to have high quality predictions. Athletes are expected to regularly sync their data through wearable devices and apps. Additionally, all user data are handled in compliance with privacy regulations (e.g. GDPR), ensuring secure and anonymized storage.

6 Methodology

6.1 Problem statement

The problem is framed as a supervised regression task, where the goal is to predict the number of calories a cyclist consumes during his ride. The output is a numerical value (calories), while the input features are body metrics and historical ride details. More details about the dataset are reported in Section 6.2.

6.2 Data

The model is trained using cycling data sourced from the various social networks designed for tracking and sharing fitness activities, such as Garmin Connect, Strava, and TrainingPeaks, as well as from wearable devices. On these platforms, athletes can share their everyday trainings and the data collecting from these sources is trivial, provided that a user has given permission for that.

The model utilizes features from three distinct context areas: body metrics, historical ride data, and the specific training plan.

Body metrics

The body metrics section outlines the key physical attributes of the athlete, as shown in Table 3.

Label	Type	Description
age	numeric	The age of the athlete.
gender	categorical	The gender of the athlete is female or male.
weight	numeric	The weight of the athlete.
height	numeric	The weight of the athlete.
fitness_level	categorical	The level of the athlete is beginner or intermediate or professional.
sleep_tracking	time	Hours of sleep during a night [HH:mm].
avg_hr_day	numeric	The average heart rate throughout the day up until that moment, so before the ride, excluding sports activities.
stress_level	numeric	The stress level range that a person has during the day.
body_battery	numeric	It identifies meaningful physiological states and it describes the impact they have on the body's energy levels.
calories_burned	numeric	The total calories burned up until that moment, so before the ride.

Table 3: Body Metrics Data

For the following variables, collected by wearable devices, it is necessary to explain better what they are:

- **Sleep tracking**: sleep is crucial to the physical and mental well-being. Regularly getting enough quality sleep promotes good health, can improve the mood, helps maximize the benefits of exercise and provides many additional benefits [5].
- **Stress level**: the wearable device analyzes the heart rate variability while the athlete is inactive to determine his overall stress. Training, physical activity, sleep, nutrition, and general life stress all impact in the stress level. Knowing the stress level can help to identify stressful moments throughout the day. The stress level range is from 0 to 100, where 0 to 25 is a resting state, 26 to 50 is low stress, 51 to 75 is medium stress, and 76 to 100 is a high stress state [6].
- **Body battery**: the body battery feature works by continuously analyzing combinations of heart rate, heart rate variability (HRV) and movement data while the athlete wears the

smartwatch. The goal of this analysis is to identify meaningful physiological states and to describe the impact they have on the body's energy levels [7].

Historical ride

The historical ride section provides details about the athlete's past training sessions, as shown in Table 4.

Label	Type	Description
distance	numeric	The total distance covered during the ride.
moving_time	time	The actual time spent in motion during the ride with the format [HH:mm:ss].
elevation_gain	numeric	The total elevation gained during the ride.
avg_speed	numeric	The average speed achieved during the ride in miles per hour.
watts	numeric	The power to maintain during the ride.
avg_watts	numeric	The average power output during the ride.
avg_heart_rate	numeric	The average heart rate during the ride in beats per minute.
avg_cadence	numeric	The average revolutions per minute of pedals during a ride.
date	date	The date of the ride with the format [yyyy-MM-dd].
time	time	The time of the starting ride with the format [HH:mm:ss].

Table 4: Historical Ride Data

Training plan

Since the model is designed to calculate the calories to consume during a workout, it is also necessary to have the training plan to identify the type of effort involved. The training plan - created by the coach - will include, for example, the description, purpose, and target wattage for the training. In general, the model will use the different zones of the wattage that the athlete needs to maintain during the ride for a specific duration, rather than focusing on the actual workout time.

6.3 Techniques

In this section, it is reported the flow of the work using for reaching the final goal, also described in Section 7.

Data collection: as reported in Section 6.2, data and information are gathered from open-source platform storing personal and athletic activities. This is the first step of the study to create a database.

Data processing and cleaning: the second step involves data processing and cleaning, with the goal of preparing suitable data for use in the model. Several key actions are taken to clean the initial dataset and produce the final version. For instance, missing values are managed by imputation [8], while certain values, such as '*distance*', '*avg_speed*', '*avg_heart_rate*', and '*avg_cadence*', are dropped from the dataset because they are irrelevant to the outcome or introduce noise (e.g. outliers).

In Table 5, the transformations applied to the different variables are shown.

Variable	Transformation
gender	From categorical variable to integer: 0 for female, 1 for male.
fitness_level	From categorical variable to integer: 0 for beginner, 1 for intermediate, 2 for amateur, 3 for professional.
sleep_tracking	From [HH:mm] to [HH].
moving_time	From [HH:mm:ss] to [HH].
time_day	Extract the time of the starting ride from [HH:mm:ss] to the difference part of a day: 0 for the morning activity (5AM to 11AM), 1 for the afternoon activity (12 AM to 5PM), 2 for the evening activity (6PM to 11PM).
day_week	From the column '*date*' [yyyy-MM-dd] extract the day of the week [dd] of the ride (Monday=0, Sunday=6).
month	From the column '*date*' [yyyy-MM-dd] extract the month [MM] of the ride (January=0, December=11).
year	From the column '*date*' [yyyy-MM-dd] extract the year [yyyy] of the ride.

Table 5: Data processing

Features: the final dataset with the name of the features used as input data is reported in the following Table 6. While, the target variable is 'calorie_prediction'.

Feature	Type
age	integer
gender	integer
weight	float
height	float
fitness_level	integer
sleep_tracking	float
avg_hr_day	float
stress_level	float
body_battery	float
calories_burned	float
moving_time	float
elevation_gain	float
watts	float
avg_watts	float
time_day	integer
day_week	integer
month	integer

Table 6: Features

Model: the desired outcome is a Random Forest model which predicts a cyclist's ride caloric needs based on the features described. In particular, the combination of using Scikit-Learn [9] and Random Forest [10] has allowed to make the most of the potential of both tools. In fact, on the one hand, Scikit-Learn simplifies the model development process thanks to its wide range of integrated functions. While, on the other hand, Random Forest is built explores the advantageous use of regression algorithms in predicting burned calories. The advantage lies in its potential for precise calorie estimation, beneficial for fitness applications [4, 11]. In general, a Random Forest is like a team of decision trees each having slightly different specialties, and the final prediction is like a team decision made after considering everyone's input. This approach helps tackle more complex problems that a single decision tree might struggle with [12].

6.4 Experimentation and Validation

After creating the model, the first step is to split the data into a training set and a test set [13]. This involves the choice to divide the final dataset into a 80% training set and a 20% test set. Cross-validation will be employed to avoid overfitting and ensure that the model generalizes well to unseen data.

The model will be validated offline using standard regression metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate prediction accuracy.

6.5 Human-in-the-loop

Athletes will be able to provide feedback on whether the calorie recommendations were accurate after completing their rides. This feedback can be incorporated into future model updates to improve personalizing. User-specific adjustments will be applied based on repeated patterns, allowing the model to adapt to individual needs over time.

7 Implementation

High-level design

The calorie estimation model operates in a cloud-based environment, using user data (e.g., body metrics, historical ride data from different platforms, and training plans) with also an integration with wearable devices (e.g. Garmin). The system pre-process the data, pass it to the Random Forest model, and return calorie recommendations to users in real-time.

A high-level data flow diagram [14] is illustrated in the Figure 1:

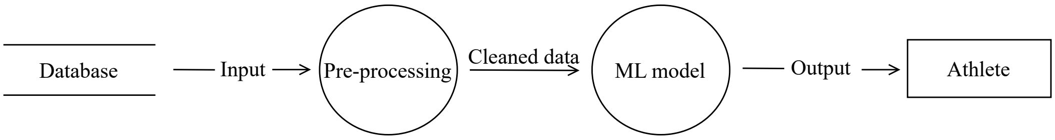

Figure 1: Data Flow diagram

Infrastructure

The system will be hosted by Amazon Web Services (AWS) which is a cloud-based infrastructure that ensures scalability and reliability. Cloud hosting provides flexibility and easy integration with external services.

Performance

To meet throughput and latency requirements, the system will based on cloud-based auto-scaling features to handle increased loads. Horizontal scaling will employed: in fact, this ensures that more instances of the service are spun up as demand increases.

Security

The model will based on sensitive data, so the security of the service is a top priority. In fact, it will be protected by a firewall and all API endpoints will be secured with HTTPS, ensuring encrypted communication between clients and the server.

Data privacy

To ensure user data privacy, the system will comply with GDPR regulations. Sensitive user information such as body metrics, historical ride data, and training plans will be encrypted.

Monitoring & Alarms

In addition to use AWS as a cloud infrastructure, the idea is also to monitor the system performance using tools such as AWS CloudWatch which is a service that monitors applications, responds to performance changes, optimizes resource usage, and provides operational status information. By collecting data across AWS resources, CloudWatch monitors critical thresholds, such as high latency or failed predictions, and triggers automatic notifications to the operations team.

Cost

The estimated monthly cost for the system is up to 100.000 NOK/month based on the following resources:

- EC2 instances for hosting and scaling the model;
- S3 storage for user data;
- RDS for structured database queries and storage;
- CloudWatch for monitoring and alerting services.

This estimate is subject to change based on factors such as the size of the dataset, the number of requests, the number of users, and overall system usage [15].

Integration points

The idea is to integrate the model with external wearable APIs (e.g., Garmin, Strava, TrainingPeaks) to pull ride data directly into the model. On the output side, it will expose a REST API for downstream clients (mobile apps, web applications) to retrieve personalized calorie recommendations. This enables easy integration with third-party cycling or fitness apps.

Risks & Uncertainties

There will always be risks and uncertainties with a system like this, including data quality, privacy, and security concerns. However, the IT team will work to minimize these issues as much as possible.

8 Conclusion

Nowadays, monitoring calorie burn remains a challenge, particularly in endurance sports, where avoiding the phenomenon known as '*bonking*' is critical. This document proposes a model that integrates with major sports platforms to address this issue. While many studies in sports focus in clustering similar athlete behaviors during performance, but not taking into consideration the nutritional aspects [16], this research aims to fill that gap by simultaneously considering fitness levels, eating habits, and lifestyle factors to provide more accurate calorie estimates.

Bibliography

[1] Juliana Exel and Peter Dabnichki. 'Precision Sports Science: What Is Next for Data Analytics for Athlete Performance and Well-Being Optimization?' In: *Applied Sciences* 14.8 (2024). ISSN: 2076-3417. DOI: 10.3390/app14083361. URL: https://www.mdpi.com/2076-3417/14/8/3361.

[2] Brandon W. *Decision Tree, Random Forest, and XGBoost: An Exploration Into the Heart of Machine Learning.* 2020. URL: https://medium.com/@brandon93.w/decision-tree-random-forest-and-xgboost-an-exploration-into-the-heart-of-machine-learning-90dc212f4948.

[3] Amol Kadam et al. 'Calories Burned Prediction Using Machine Learning'. In: *2023 6th International Conference on Contemporary Computing and Informatics (IC3I).* Vol. 6. IEEE. 2023, pp. 1712–1717.

[4] Harry Paul et al. 'CycleFit: An Analysis of Regression Models for Caloric Expenditure Prediction in Cycling Activities'. In: *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI).* Vol. 2. IEEE. 2024, pp. 1–6.

[5] Garmin. *Sleep Tracking: Advanced Sleep Monitoring with Garmin Wearables.* 2024. URL: https://www.garmin.com/en-US/garmin-technology/health-science/sleep-tracking/.

[6] Garmin Support. *How Is Stress Level Measured on My Garmin Watch?* Accessed: 2024-10-06. 2024. URL: https://support.garmin.com/en-US/?faq=WT9BmhjacO4ZpxbCc0EKn9 (visited on 06/10/2024).

[7] Garmin. *Body Battery: Energy Monitoring Technology.* 2024. URL: https://www.garmin.com/en-US/garmin-technology/health-science/body-battery/.

[8] scikit-learn. *sklearn.impute.IterativeImputer.* 2024. URL: https://scikit-learn.org/1.5/modules/generated/sklearn.impute.IterativeImputer.html.

[9] Fabian Pedregosa et al. 'Scikit-learn: Machine learning in Python'. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[10] Hoss Belyadi and Alireza Haghighat. 'Chapter 5 - Supervised learning'. In: *Machine Learning Guide for Oil and Gas Using Python.* Ed. by Hoss Belyadi and Alireza Haghighat. Gulf Professional Publishing, 2021, pp. 169–295. ISBN: 978-0-12-821929-4. DOI: https://doi.org/10.1016/B978-0-12-821929-4.00004-4. URL: https://www.sciencedirect.com/science/article/pii/B9780128219294000044.

[11] Marte Nipas et al. 'Burned calories prediction using supervised machine learning: Regression algorithm'. In: *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T).* IEEE. 2022, pp. 1–4.

[12] Brandon Wong. *Decision Tree, Random Forest, and XGBoost: An Exploration into the Heart of Machine Learning.* 2024. URL: https://medium.com/@brandon93.w/decision-tree-random-forest-and-xgboost-an-exploration-into-the-heart-of-machine-learning-90dc212f4948.

[13] Rashida Nasrin Sultana Khurana. *Train, Validation, and Test Sets.* 2024. URL: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7.

[14] Wikipedia contributors. *Data-flow diagram.* 2024. URL: https://en.wikipedia.org/wiki/Data-flow_diagram.

[15] Amazon Web Services. *How AWS Pricing Works.* 2024. URL: https://docs.aws.amazon.com/pdfs/whitepapers/latest/how-aws-pricing-works/how-aws-pricing-works.pdf.

[16] Antonios Pantazopoulos and Manolis Maragoudakis. 'Sports & nutrition data science using gradient boosting machines'. In: *Proceedings of the 10th Hellenic Conference on Artificial Intelligence.* 2018, pp. 1–7.

[17] Tianqi Chen and Carlos Guestrin. 'Xgboost: A scalable tree boosting system'. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.* 2016, pp. 785–794.

Appendix

A Alternatives

One of the possible alternative which it is taken into consideration, was to use XGBoost model instead of Random Forest. XGBoost, which stands for '*eXtreme Gradient Boosting*', is an advanced implementation of the gradient boosting algorithm. Gradient boosting is a machine learning technique where the main idea is to combine many simple models, also known as '*weak learners*', to create an ensemble model that is better at prediction [17]. However, XGBoost has some disadvantages. It is prone to overfitting if not properly tuned, requiring careful adjustment of parameters like learning rate, tree depth, and the number of trees. Additionally, finding the optimal hyperparameter settings can be challenging, often necessitating grid or randomized search methods [12].

B Glossary

API Application Programming Interface.

AWS Amazon Web Services.

EC2 Elastic Compute Cloud.

GDPR General Data Protection Regulation.

HR Heart Rate.

HRV Heart Rate Variability.

HTTPS HyperText Transfer Protocol Secure.

MAE Mean Absolute Error.

PII Personally Identifiable Information.

RDS Relational Database Service.

RMSE Root Mean Squared Error.

S3 Simple Storage Service.

XGB eXtreme Gradient Boosting.