Calorie Expenditure Prediction with Machine Learning

Miles Anderson

University of Oregon

Mander20@uoregon.edu

Abstract – As the number of overweight Americans continues to increase it is increasingly important to prioritize proper weight management. To accomplish this one must be aware of the number of calories they are consuming and burning in a day. While tracking caloric intake is somewhat straightforward, tracking expenditure is more difficult. Thankfully, with the advent of smartphones and watches the data necessary to calculate calorie expenditure is now available. This paper analyzes two regression models', decision trees and MLPs, and their ability to predict calories burnt in a given exercise. The data used in this experiment was procured from the "Predict Calorie Expenditure" Kaggle competition. A variety of hyperparameters were tuned for both models and the result of that tuning is analyzed. Furthermore, learning curves, feature importance, and training time were captured. This experiment indicated that while MLPs took longer to train they were slightly more accurate than decision trees at predicting calorie expenditure.

I Introduction

Weight and health are inextricably related. With nearly 40% of Americans classified as obese in August 2023 it has become a national health crisis [2]. Obesity is related to a long list of health issues including diabetes, heart disease, cancer, and even mental health issues [3]. As a result, emphasis on proper health and weight management are more important than ever. When it comes to weight management it boils down to the difference between the number of calories one takes in, and the amount they burn off [1]. With this being the most important factor in any weight loss program, it is important people be able to track the number of calories they burn.

With an influx of individual health and medical data due to smartphones and watches, tracking calories burnt has never been more accessible. Important metrics can be captured through these devices such as duration of a workout, heart rate, and body temperature. With the addition of information such as sex, age, height, and weight an accurate estimate can be drawn of how many calories were expended during a given workout. This prediction can serve as a helpful tool for people aiming to lose weight, allowing them to more accurately track their calories burnt, and adjust their caloric intake accordingly. To make accurate predictions on this data, machine learning can be utilized.

This paper analyzes the accuracy and efficiency of two ML regressors. This will be done using data from the Kaggle competition "Predict Calorie Expenditure". This data was chosen for two main reasons. One, its cleanliness, which allowed for greater focus on training and tuning. And two, its size. The data consists of 750,000 entries, making it possible to split into many subgroups for hyperparameter tuning. There are seven features: sex, age, height, weight, duration, heart rate, and body temperature. These make it an ideal dataset for working with regression models.

This paper will focus its analysis on decision trees, and multi-layer perceptrons (MLP). The decision tree has the advantage of being simple and straightforward. It serves as a baseline. It also allows for a measurement of feature importance, shedding light on which factors contribute most to the number of calories burned in a workout. The MLP acts as a foil to the decision tree with its ability to capture complex nonlinear patterns, finding subtleties a shallow decision tree might miss. By evaluating both models on identical data splits and metrics (MAE and RMSLE) they can be directly compared, and it can be said whether the MLP is worth the higher computational cost.

The rest of the paper is organized as follows. Section two reviews past research on calorie expenditure prediction. Section three details methodologies and the steps taken to carry out this experiment. Section 4 explores the results of the experiment, drawing comparisons between the two models and analyzing the affects their hyperparameters had on their performance. Finally, in section five conclusions will be drawn and outlines for future research will be laid.

II Background

A. *Literature Review*

[4] This research article aimed to show the effectiveness of a Random Forest Regressor in predicting calorie expenditure. The authors used 15,000 data points with features such as age, gender, height, weight, duration, heart rate, and body temperature. After cleaning and exploring the data, it was used to train a Random Forest Regressor. Hyperparameters such as number of trees, max depth, and max features were carefully considered. The authors designed a web application to be used alongside their trained model. The model had a Root Mean Squared Error of 8.3.

[5] The research in this article aims to find accurate calorie-burn estimates by testing various regression models. The dataset consists of 15,000 data points with eight features including age, height, weight, duration, heart rate, body temperature, metadata, and gender. The models this data was applied to was linear regression, ridge regression, and random forest regression. The best performing model was random forest regression achieving scores of 8.13 on MSE, 2.85 on RMSE, and 1.81 on MAE.

B. *Models*

Regression models were used in this research to capture a continuous outcome on nonnegative targets. The specific models to be analyzed in this paper are decision trees and MLPs. Performance will be assessed using mean absolute error (MAE) and root mean squared logarithmic error (RMSLE). MAE was chosen to measure the average absolute difference between the predicted and the true values. RMSLE was chosen to provide a percent-error, and to provide a correlation to the original Kaggle competition where the data was pulled from. To avoid overfitting k-fold cross validation was used to train the models and tune their hyperparameters. This framework lays the foundation for further comparison of decision trees and multilayer-perceptrons on the calories burnt prediction problem.

The key hyperparameters analyzed in this research for the decision tree was max depth, minimum sample leaves, minimum sample split, and max features. The advantage to the decision tree is its transparency. This transparency allowed for the extraction of feature-importance. The pitfall of the decision tree is its lack of flexibility, which is why the decision tree acts as a baseline to be compared against the more flexible MLP.

The key hyperparameters examined for MLP are hidden layer sizes, alpha (for L2 regularization), learning rate initializer, and max iterations. The advantage of the MLP is its ability to pick up on subtle patterns in the data a simpler model might miss. Its drawback is its expensive computation.

III Methods

All the programming was done using python 3.11.5 and sickit-learn 1.x.

A. *Data Source and Preprocessing*

The data for this experiment was taken from the "Predict Calorie Expenditure" Kaggle competition. It contained 750,000 entries and the following features: sex, age, height, duration, heart rate, body temperature, and calories burned.

The train.cv file was read into a Pandas data frame. It was confirmed that there were zero missing values, and the immutable column 'id' was dropped. The sex column, being of type object, was one-hot encoded into a binary integer 'sex_male' column. For the MLP z-score standardization was applied using 'StandardScaler'. No scaling was needed for the decision tree. The data was split 80/20. The 80% was for training/validation and the other 20% was used for testing. For reproducibility a random state of 42 was used.

B. *Evaluation Metrics and Validation Strategy*

The two metrics used were mean absolute error (MAE) and root mean squared logarithmic error (RMSLE). MAE captured the average absolute deviation and was the primary tuning tool.

RMSLE captured relative error on a log scale and was only computed once on the final test predictions. A 5-fold cross-validation was used on the 80% of the data set aside for training and validation.

C. *Decision Tree Regressor*

The exact model used in the implementation for the decision tree was 'DecisionTreeRegressor' and came from sickit-learn. The grid was set over the following hyperparameters:

- max_depth: {None, 3, 5, 8, 12}
- min_samples_leaf: {1, 2, 5, 10}
- min_samples_split: {2, 5, 10}
- max_features: {None, 'sqrt', 'log2'}

Each hyperparameter combination was fitted via 5-fold CV and the mean negative MAE was recorded. The combination of hyperparameters with the highest mean negative MAE was chosen. The best tree was then retrained on the full 80% and then predicted on the 20% set aside for testing for a final MAE and RMSLE score.

D. *Multilayer Perceptron Regressor*

First a pipeline was built to facilitate the MLP. Then the grid was set over the following hyperparameters:

- mlp__hidden_layer_sizes: {(50,), (100,), (50, 50), (100, 50)}
- mlp__alpha: {1e-4, 1e-3, 1e-2}
- mlp__learning_rate_init: {1e-3, 1e-4}
- mlp__max_iter: {300, 500}

5-fold CV with 'GridSearchCV(scoring='neg_mean_absolute_error')' was used on the pipeline. Early stopping was implemented for when validation loss failed to improve over 10 epochs. The best pipeline was refit on the full training plus validation split and a prediction is made on the test split to give a final MAE and RMSLE score.

E. *Diagnostic Plots and Timing*

For both models a feature-importance bar chart is generated. The decision tree graph gives its results from the final estimator. The MLP gives permutation importance on the 80% using 'permutation_importance()'. Then a hyperparameter validation curve is generated for both models where their key hyperparameters are plotted against the MAE score. For the decision tree the key parameter is max_depth, and for the MLP it is hidden-layer size. Afterwards a learning curve is generated for both models. Finally, both models are timed on how long it takes to tune the hyperparameters and how long it takes to do the final test.

IV Experiments

A. *Setup*

This experiment was done using python 3.11 and sickit-learn. Both models are split using a fixed random state of 42. The split used on the data was 80/20 train and validation/test. To clean up the data get rid of the id column and do one-hot encoding on the sex column. Finally do z-score scaling for the MLP.

B. *Hyperparameter Tuning*

B.1 *Decision Tree*

- Max Depth: {None, 3, 5, 8, 12}
  - This hyperparameter was chosen along with its range to show whether a simpler or more complex tree would be better for this problem.
- Minimum Samples Leaf: {1, 2, 5, 10}
  - This hyperparameter along with its range tests how smooth the model should be for this problem.
- Minimum Samples Split: {2, 5, 10}
  - This hyperparameter was chosen to see the effect variance had on the model.
- Max Features: {None, 'sqrt', 'log2'}
  - The max features parameter improves generalization helping to round out the decision tree.

Together these hyperparameters and their ranges allow for the tree to either be very simple or very complex. This range makes it possible to home in on the sweet spot where the tree is neither too complex nor too simple.

B.2 *MLP*

- Hidden Layer Sizes: {(50,), (100,), (50, 50), (100, 50)}
  - The hidden layer hyperparameter allows for adjustment of the model's complexity.
- Alpha (L2 regularizer): {1e-4, 1e-3, 1e-2}
  - This parameter was used to balance regularization in the model.
- Learning Rate: {1e-3, 1e-4}
  - This hyperparameter shows whether the model would benefit from a lower validation loss.
- Max Iterations: {300, 500}
  - This hyperparameter is used in junction with the early stop to halt training when validation loss doesn't improve the model anymore.

This mix of hyperparameters and their ranges allows for a balancing act between underfitting, overfitting, and computing cost.

The hyperparameters for both models were picked purposely to mirror each other. The max depth parameter and the hidden layer size parameter affect the capacity of their respective models. The minimum sample leaf and alpha value control the smoothness of the models. The minimum sample split and learning rate initializer control feature sampling. And the maximum iterations for the MLP as well as a combination of the max depth and leaf/sample parameters curb overfitting in their respective models. This way it is a direct and fair comparison between the decision tree and the MLP.

B.3 *Comparison*

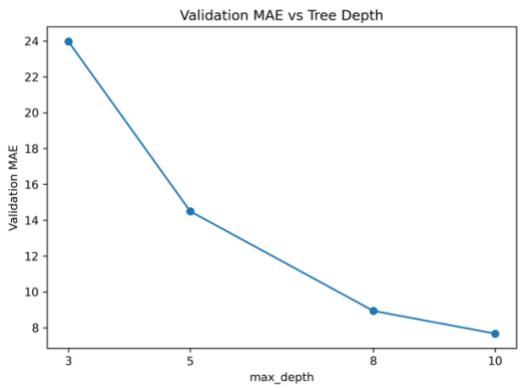
Best Hyperparameters for Decision Tree:

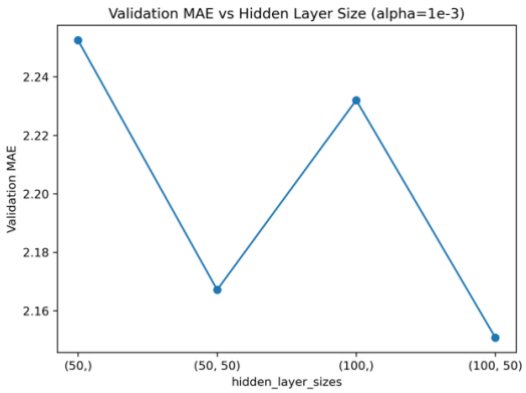
Criterion	Max Depth	Max Features	Minimum Sample Leaves	Minimum Sample Split	Test MAE
Squared Error	None	None	10	2	2.47

Best Hyperparameters for MLP:

Alpha	Hidden Layer Size	Learning Rate	Max Iterations	Test MAE
.01	(100, 50)	1e-4	300	2.14

These two tables show that while the inner workings of both models are quite different, they have managed to converge on performance. The decision tree is large and smooth due to its lack of maximum depth and large minimum sample leaves. The MLP uses dense continuous transformations over only two layers. With their respective strategies both models managed to get under a 2.5 MAE. This difference in approach can be further visualized using these graphs.



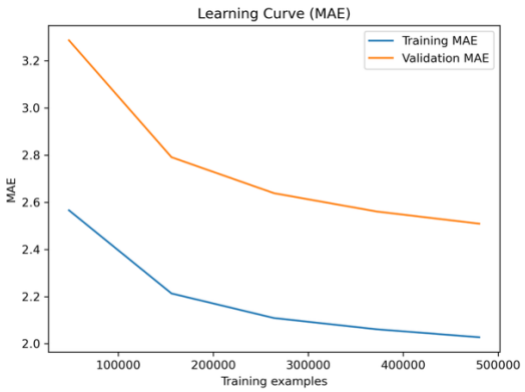


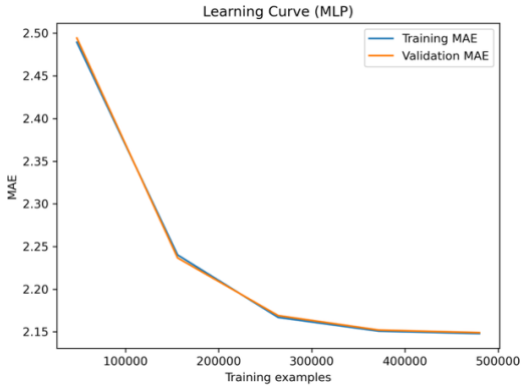
On the left is the validation MAE vs. the max depth of the decision tree (with minimum sample leaves of 10, minimum sample split of 2, and max features equaling None). It shows MAE falling sharply from 3-8 then flattening out. This indicates that once the tree reaches a certain depth and sufficient branching is allowed more depth has little benefit.

The graph on the right shows the validation MAE vs. hidden layer sizes (with alpha equaling 1e-3, learning rate initializer being 1e-3, and max iterations being 300). The two layer network of (100,50) giving the best MAE shows that moderate depth paired with good regularization yields the best trade-off between the model's capacity and overfitting.

The two graphs show a different approach to finding the best tuning. The decision trees graph is monotonic, and the tree clearly benefits from more depth. The MLPs graph is non-monotonic and while it does benefit from more depth, it must balance it with regularization.

C. *Learning Curve Analysis*



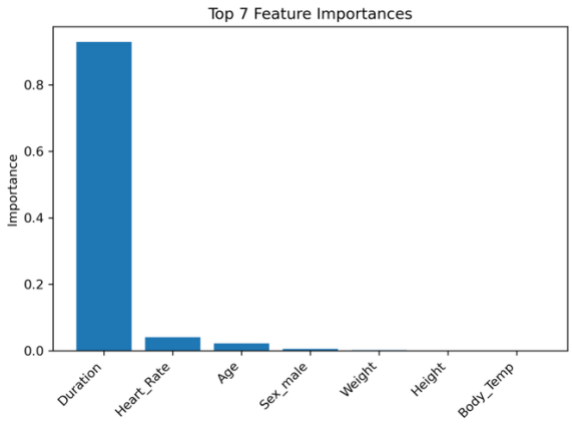


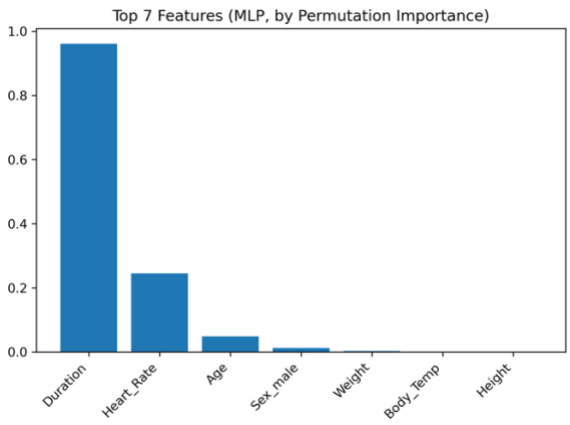
These graphs show the learning curves of both models (the decision tree being on the left and the MLP being on the right). The graphs show how the percentage of the training data being used affects the MAE.

The decision tree's training MAE is much lower than its validation MAE even when it's seen very little data. This is indicative of the decision trees tendency to overfit on small datasets. The gap seems to get a little better as the amount of data increass, signifying more data could improve the performance of the decision tree.

In contrast there is almost no gap between the MLPs training and validation MAE for the entire duration of training. This suggests there is little under or overfitting in the MLP model. The MLP also improves it's MAE score faster than the decision tree and starts with a better MAE. This shows that the MLP is better suited for smaller datasets than the decision tree.

D. *Feature Importance*





The left graph shows the importance of each feature in making predictions for the decision tree. The tree's built-in 'feature_importances_' was used. The graph on the right displays permutation importances for the MLP. According to both graphs duration is the most critical feature by a wide margin. However, it is worth noting the importance of the heart rate and age features are significantly higher for the MLP. This is because the MLP is a more sensitive model than the decision tree. This sensitivity could be a reason the MLP achieves a lower MAE.

E. *Timing*

Both tuning and training time and inference time were computed on the models using 'time.perf_counter()'.

Model	Tuning & Training Time (s)	Inference time
Decision Tree	74.9	0.053
MLP	468.6	0.150

The decision tree is significantly faster than the MLP. It's tuning and training time was nearly 16% faster and its inference time was almost 35% faster. This poses a clear tradeoff between the two models. In any setting where speed is important the decision tree is more well suited. In a setting where time doesn't matter and high accuracy is needed the MLP might be better.

V Conclusion

In this experiment two regression models, decision trees and MLPs, were compared side-by-side on a calorie expenditure prediction problem. An 80/20 train/test split was used on a set of 750,000 datapoints. These splits alongside 5-fold CV were used to train various hyperparameters on MAE. The hyperparameters showed a slight preference for complexity in the decision tree and a strong balance between complexity and regularization for the MLP. Feature importance was also analyzed showing that Duration is by far the most crucial. With there being differences in the importance of heart rate and age for the two models. It was found that while the MLP was

slightly more accurate it took significantly longer to train. Many tradeoffs exist between the two models. In a setting where there is limited data or time is not a concern the MLP might be a better choice. On the other hand, if speed and computation cost is important the decision tree is probably the better model.

Future research could be extended to other models, in particular random forests. Many existing discussions of machine learning being used to predict calorie expenditure explore random forests. In addition, these random forests often outperform their counterparts. Comparing them to MLPs could provide further insight on the pros and cons of using an MLP on this problem.

References:

[1] *Counting calories: Get back to weight-loss basics*. (n.d.). Mayo Clinic. https://www.mayoclinic.org/healthy-lifestyle/weight-loss/in-depth/calories/art-20048065

[2] Emmerich, S., Fryar, C., Stierman, B., & Ogden, C. (2024). Obesity and Severe Obesity Prevalence in Adults: United States, August 2021–August 2023. *CDC*. https://doi.org/10.15620/cdc/159281

[3] *Health Risks of Overweight & Obesity*. (2025, January 31). National Institute of Diabetes and Digestive and Kidney Diseases. https://www.niddk.nih.gov/health-information/weight-management/adult-overweight-obesity/health-risks

[4] Kadam, A., Shrivastava, A., Pawar, S. K., Patil, V. H., Michaelson, J., & Singh, A. (2023, September). Calories burned prediction using machine learning. In *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)* (Vol. 6, pp. 1712-1717). IEEE.

[5] M. Nipas, A. G. Acoba, J. N. Mindoro, M. A. F. Malbog, J. A. B. Susa and J. S. Gulmatico, "Burned Calories Prediction using Supervised Machine Learning: Regression Algorithm," 2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T), Raipur, India, 2022, pp. 1-4, doi: 10.1109/ICPC2T53885.2022.9776710. keywords: {Training;Machine learning algorithms;Computational modeling;Data visualization;Predictive models;Prediction algorithms;Cleaning;calories;regression model;prediction;machine learning},