CALORIES BURN PREDICTION

Abstract

This report explores the application of multiple machine learning algorithms—Linear Regression, Ridge Regression, Lasso Regression, Artificial Neural Network (ANN), Random Forest Regressor, and XGBoost Regressor—on a dataset aimed at predicting calorie consumption based on various food-related features. The dataset was preprocessed by handling missing values and normalizing features to ensure consistency across the data. Multiple models were employed to predict calorie values, and their performance was evaluated using key metrics including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared, and Mean Absolute Error (MAE).

Datasets

	Data Set Characteristics	Attribute Characteristics	Associated Tasks	Number of Instances	Number of Attributes
Dataset	Calorie Burnt Prediction	Real	Regression	15000	9

Why I find Datasets Interesting

The Calorie Burn Prediction dataset is interesting because it focuses on predicting the number of calories burned based on various physical activities and personal characteristics. This is highly relevant in the fields of health and fitness, where accurate predictions can help individuals track and manage their physical activity and health goals. By analysing this dataset, patterns that contribute to effective calorie burning can be identified, allowing for personalized fitness plans or recommendations. This could assist in promoting healthier lifestyles, improving fitness routines, and even preventing lifestyle-related health issues, making the dataset valuable for the health and wellness industry.

Data Characteristics

The Calorie Burn Prediction dataset consists of **15,000 records** and **9 features**. These features include various personal characteristics (such as age, weight, height) and activity-related data (like duration of exercise, type of exercise, and intensity). Most of the columns are numerical. The target variable, 'Calories', is a continuous value representing the number of calories burned during an activity.

The dataset contains no missing or null values, and any outliers have been managed during preprocessing. Key preprocessing steps include normalization of numerical features to ensure that they are on a comparable scale, as well as encoding categorical variables (if applicable). The data is ready for training machine learning models, with minimal preprocessing required.

Data Preparation and Preprocessing for Calorie Burnt Prediction Dataset

The Calorie Burn Prediction dataset, consisting of 15,000 records and 9 features, was loaded for analysis. The features are a mix of numerical and categorical data, with numerical features such as age, weight, height, and exercise duration, along with categorical features like exercise type. The target variable, "Calories," represents the number of calories burned during physical activity.

Initially, the structure of the dataset was checked to verify the number of records and features, along with their respective data types. The dataset consists of numerical data for most features. The target variable, "Calories," is continuous, requiring regression models for prediction.

A statistical summary of the numerical features was generated to gain insights into the dataset. Key statistics, such as mean, standard deviation, minimum, and maximum values, were calculated for variables like age, weight, height, and exercise duration.

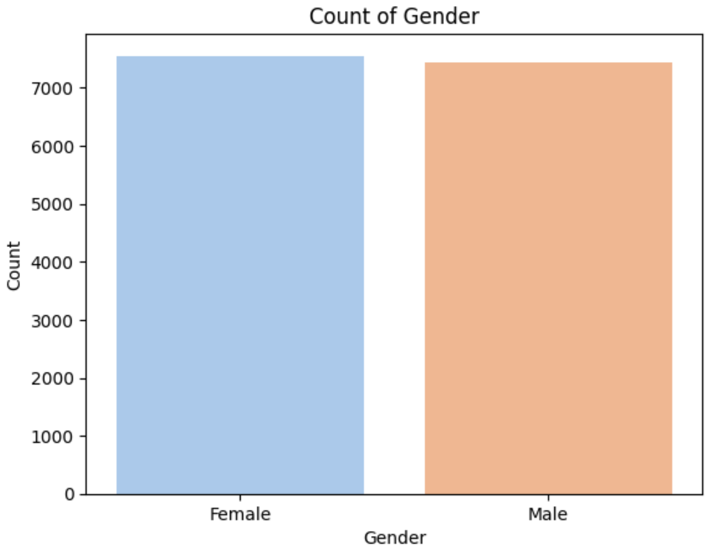
Further data quality checks were performed, confirming that the dataset contains no missing or null values, as observed through the data.info() function. Outliers were also checked, and any necessary steps were taken to handle them appropriately. Since the dataset includes both numerical and categorical features, preprocessing steps involved normalizing the numerical features to ensure they are on a similar scale and encoding the categorical features using one-hot encoding.

With no missing values and all features preprocessed, the dataset is now ready for training machine learning models to predict the number of calories burned during physical activities.

Data Visualization Calorie Burnt Prediction Data

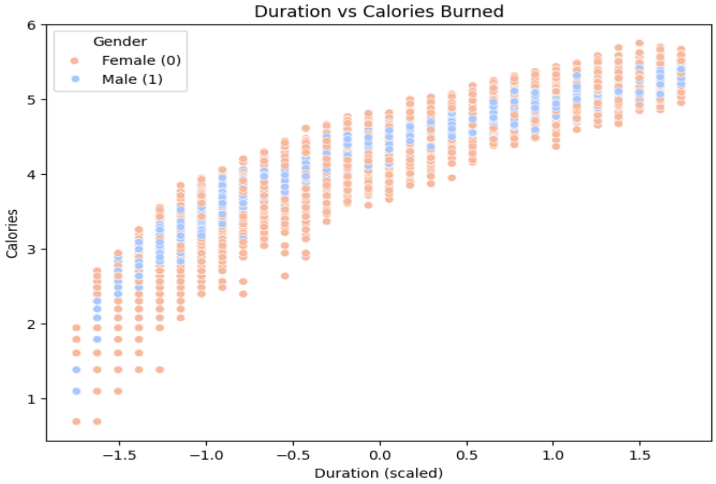
Gender Distribution Visualization

The graph shows the gender distribution within the dataset, with a significantly higher count of females compared to males.



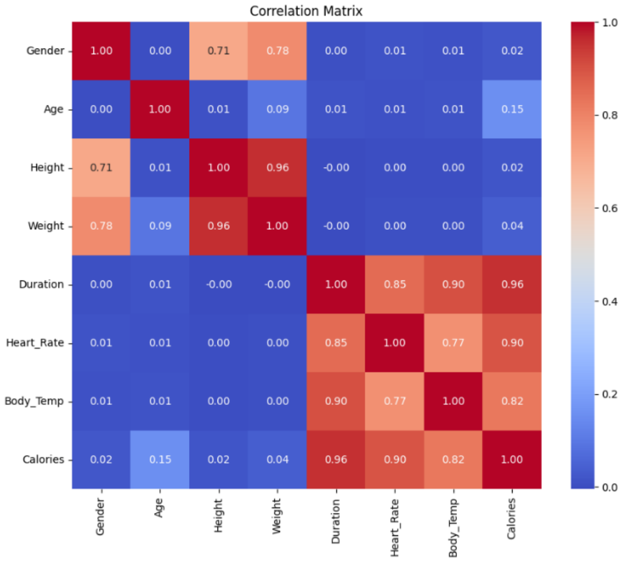
Relationship Between Duration and Calories Burned by Gender

The plot shows the relationship between the duration (scaled) and the number of calories burned, with data points separated by gender. The scatter plot reveals a positive correlation between duration and calories burned, with male participants generally burning more calories than female participants for the same duration of activity.



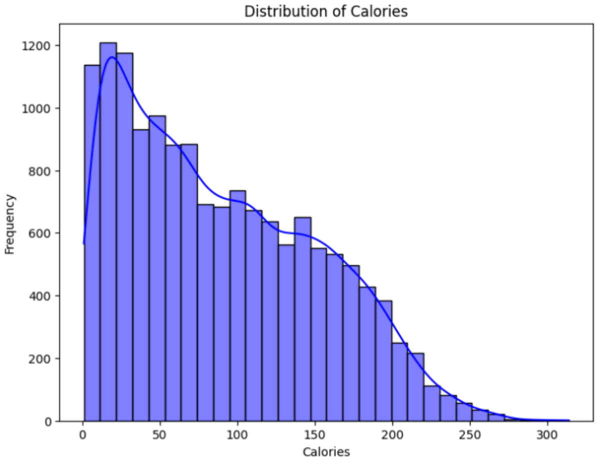
Feature Correlation Analysis in Calorie Burnt Dataset

The heatmap shows key fitness data relationships: Duration strongly correlates with Calories (0.96), Body Temperature (0.90), and Heart Rate (0.85), while Gender correlates with Weight (0.78) and Height (0.71). Height and Weight show high correlation (0.96), and Age has minimal correlation with other variables. The color intensity indicates correlation strength, from blue (0) to red (1).



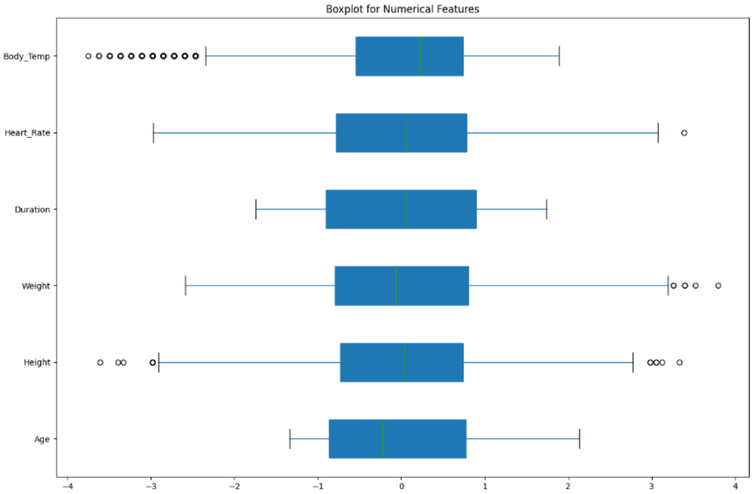
Distribution of Calories Burned

The plot displays a histogram with a smooth kernel density estimate (KDE) curve to visualize the distribution of calories burned. The histogram shows a clear right-skewed distribution, with the majority of data points concentrated in the lower calorie ranges and a long tail extending towards higher calorie values.



Visualizing the Distribution of Numerical Features Using Boxplots

The boxplot shows that body temperature has low-end outliers but consistent overall readings, while heart rate varies widely with one high outlier. Duration shows balanced spread, and both weight and height have high-end outliers. Age is symmetrically distributed with no outliers. The green lines show medians, and blue boxes represent the middle 50% of data for each metric.



Data Model Training and Evaluation

Feature Selection, Normalization, and Data Splitting for Calorie Burn Prediction Model

In this phase, relevant features were selected from the dataset for analysis. The variable X was created by removing the target variable "Calories" from the dataset, while y was defined as the "Calories" column, representing the number of calories burned during physical activity.

Next, data normalization was applied using the StandardScaler to scale the numerical features (such as age, weight, height, and exercise duration). This ensured that these features have a mean of 0 and a standard deviation of 1. Normalization is crucial for balancing the model, as it helps prevent features with larger scales from dominating the learning process, which can negatively affect the model's performance.

For categorical features like "Gender" one-hot encoding was applied to convert them into a format suitable for machine learning models, transforming the categories into binary columns.

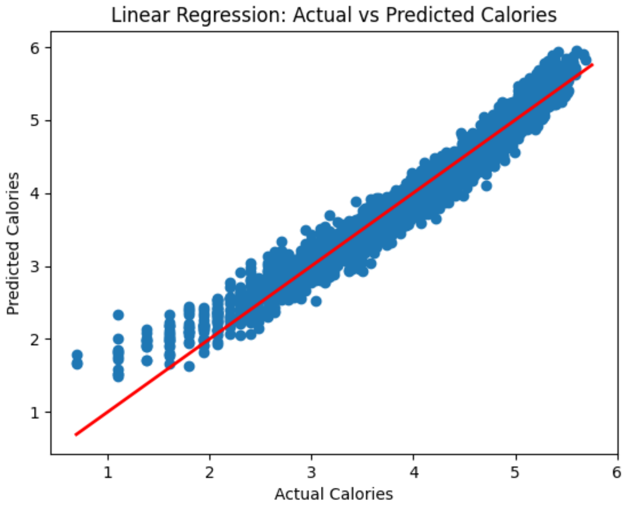
Finally, the dataset was split into training and testing sets, with 80% of the data allocated for training the model and 20% for testing. The train_test_split function was used, with stratification applied to maintain the same distribution of exercise types and other key variables in both the training and testing sets. This ensures that the model is properly evaluated on a representative portion of the dataset and avoids bias in the evaluation process. This step is essential for effectively training and testing the model to predict the calories burned based on the input features.

Linear Regression Model Initialization and Training

In this section, a Linear Regression model was initialized and trained on the dataset. Linear Regression is a fundamental machine learning algorithm used for predicting continuous values based on linear relationships between features and the target variable. The model was trained using the training data (X_train and y_train), and predictions were made on the test data (X_test). The model's performance was evaluated using key metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2). The results showed a Mean Squared Error of 0.0371, a Root Mean Squared Error of 0.1927, and an R-squared value of 0.9583. These metrics indicate that the model fits the data well and performs effectively in predicting the target variable.

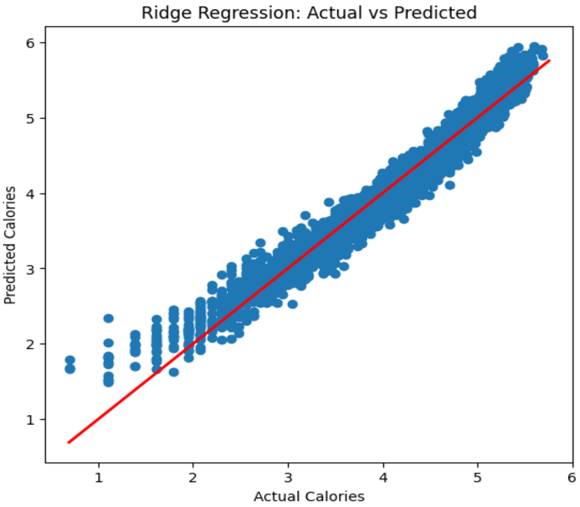
Linear Regression Model Evaluation and Comparison

The model's R-squared values for both the training and testing sets were found to be 0.9594 and 0.9583, respectively, demonstrating that the model performs consistently across both datasets. These values suggest that the Linear Regression model accounts for a significant proportion of the variance in the target variable. The performance of Linear Regression is comparable to that of Ridge Regression, with similar MSE, RMSE, and R-squared values.



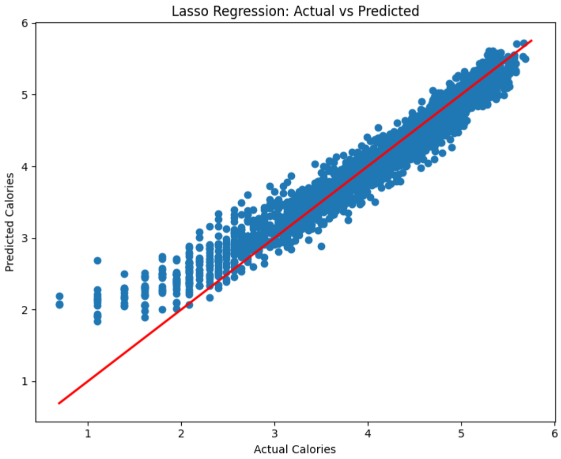
Ridge Regression Model Initialization, Training and Evaluation

In this section, a Ridge Regression model was initialized with an alpha value of 1.0, which controls the regularization strength. Ridge Regression, a form of linear regression, is useful when dealing with multicollinearity or when there are many features in the dataset, as it adds a penalty to the model to reduce overfitting. The model was trained on the training data (X_train and y_train) and then used to make predictions on the test set (X_test). The model's performance was evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared metrics. The results showed a Mean Squared Error of 0.0371, a Root Mean Squared Error of 0.1927, and an R-squared value of 0.9583, indicating a good fit and predictive power of the model. The R-squared values for the training and testing sets were 0.9594 and 0.9583, respectively, confirming that the model performs consistently across both datasets.



Lasso Regression Model Initialization, Training and Evaluation

In this section, a Lasso Regression model was initialized with an alpha value of 0.1. Lasso Regression is another type of regularized linear regression that uses L1 regularization, which helps in feature selection by shrinking some coefficients to zero. This can be especially useful when there are many irrelevant or redundant features. The model was trained on the training data (X_train and y_train) and used to predict values on the test set (X_test). The model's performance was assessed using MSE, RMSE, and R-squared metrics. The results showed a Mean Squared Error of 0.0583, a Root Mean Squared Error of 0.2414, and an R-squared value of 0.9346. The R-squared values for the training and testing sets were 0.9358 and 0.9346, respectively, indicating that the model performs well but with slightly lower accuracy compared to Ridge Regression.

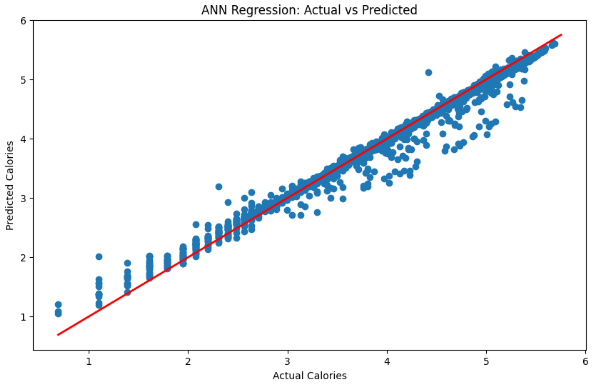


Artificial Neural Network (ANN) Model Initialization and Training

In this section, an Artificial Neural Network (ANN) was built using the Keras Sequential API. The network architecture consists of three layers: the input layer, a hidden layer with 64 units and ReLU activation, another hidden layer with 32 units and ReLU activation, and an output layer with one unit to predict the target variable, which is the calorie value. The model was compiled using the Adam optimizer, suitable for this regression task, and the mean squared error (MSE) loss function. The model was trained for 50 epochs with a batch size of 32 using the training data (X_train and y_train). During the training, the model adjusted its internal weights and biases to minimize the error, gradually improving its ability to make accurate predictions.

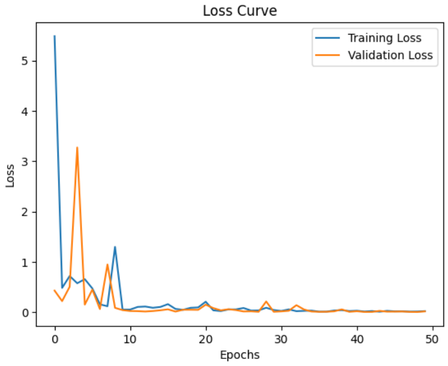
ANN Model Predictions and Evaluation

In this section, predictions were made on the test dataset (X_test) after training the ANN model. The model's performance was evaluated using metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared. The MSE, RMSE, and R-squared values for the model were calculated, yielding a MSE of 0.0148, RMSE of 0.122, and an R-squared value of 0.983. These metrics indicate that the model's predictions are fairly close to the actual values, with a good fit for the data. Additionally, the R-squared values for both training and testing data were also calculated, showing a train R-squared of 0.985 and test R-squared of 0.983, confirming the model's strong predictive capability.



ANN Model Loss Curve Visualization

The final section includes visualizations to track the training process and assess the model's learning progress. A scatter plot was created to compare the actual versus predicted calorie values on the test dataset, which provides a visual understanding of the model's performance. Additionally, the training and validation loss curves were plotted to observe how the loss function decreased over the epochs. The loss curve indicates that the model improved steadily during training, with validation loss closely tracking the training loss, which suggests good generalization to unseen data.

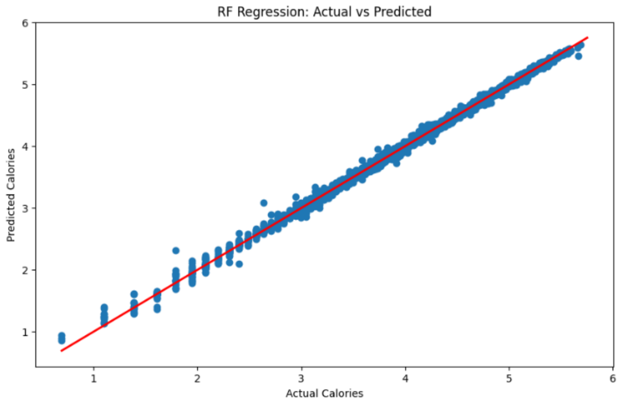


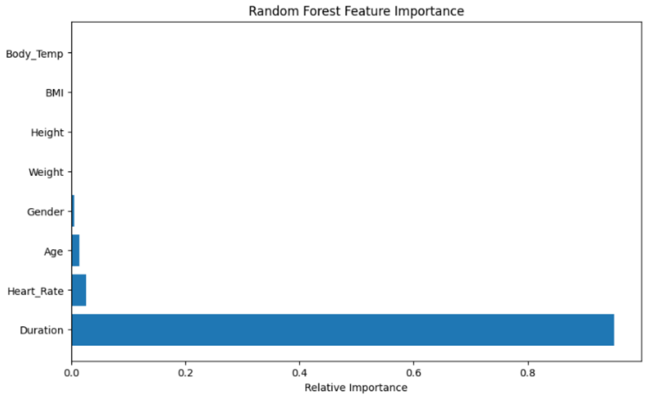
Random Forest Regressor Model Initialization and Training

In this section, a Random Forest Regressor model was initialized with 100 estimators (trees) and a random state for reproducibility. Random Forest, an ensemble learning method, works by constructing multiple decision trees and averaging their predictions to improve the model's accuracy. The model was trained using the training data (X_train and y_train). This training process allows the model to learn the relationship between the input features and the target variable (calories). The Random Forest algorithm is particularly useful for capturing non-linear relationships in the data and is less prone to overfitting compared to individual decision trees.

Random Forest Model Performance Evaluation

After training the model, predictions were made on the test dataset (X_test), and various performance metrics were calculated. The model achieved an impressive Mean Squared Error (MSE) of 0.00177, Root Mean Squared Error (RMSE) of 0.0421, Mean Absolute Error (MAE) of 0.0265, and an R-squared value of 0.998. These results indicate that the Random Forest model performed excellently, with a very low prediction error. The R-squared value of 0.998 suggests that the model explains 99.8% of the variance in the target variable. Additionally, the R-squared values for both the training and testing data were calculated: 0.9997 for the training data and 0.9980 for the test data, indicating strong performance on both sets.



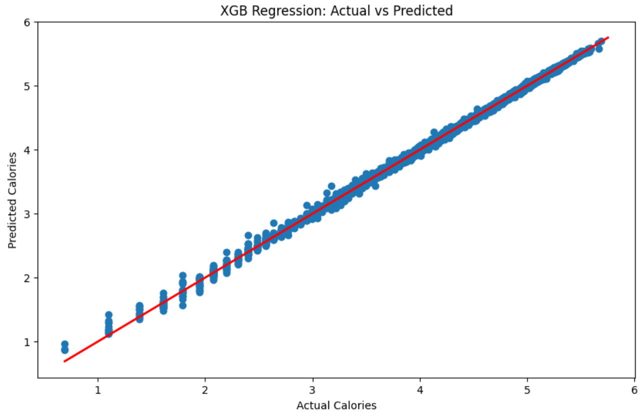


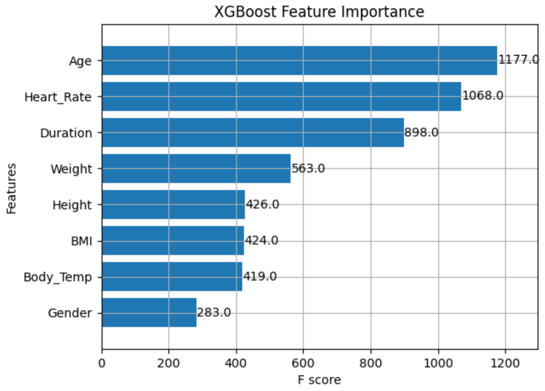
XGBoost Regressor Model Initialization and Training

In this section, an XGBoost Regressor model was initialized with the objective set to 'reg:squarederror' for regression tasks, 100 estimators (trees), and a fixed random state for reproducibility. XGBoost is a powerful gradient boosting algorithm that uses an ensemble of decision trees to make predictions. The model was trained on the training data (X_train and y_train), where it learned the relationship between the features and the target variable (calories). Gradient boosting methods, like XGBoost, are known for their high predictive accuracy and ability to handle complex relationships in the data.

XGBoost Model Performance Evaluation

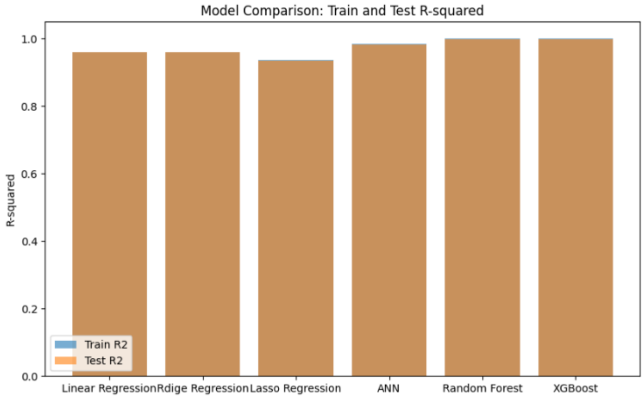
After training the XGBoost model, predictions were made on the test dataset (X_test), and several performance metrics were calculated. The model achieved a Mean Squared Error (MSE) of 0.00129, a Root Mean Squared Error (RMSE) of 0.0359, a Mean Absolute Error (MAE) of 0.0238, and an R-squared value of 0.9986. These metrics indicate strong performance, with the XGBoost model exhibiting a low prediction error and explaining 99.86% of the variance in the target variable. Additionally, the R-squared values for both the training and testing datasets were 0.9996 and 0.9986, respectively, demonstrating that the model generalizes well to unseen data.





Performance Comparison of Machine Learning Algorithms for Calorie Burnt Prediction

The performance of different machine learning algorithms for predicting calorie consumption was evaluated using various metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared. **Linear Regression** and **Ridge Regression** both performed similarly, with MSE values of 0.0371, RMSE values of 0.1927, and R-squared of 0.9583, indicating that these models explained around 95.83% of the variance. **Lasso Regression** had slightly worse performance, with an R-squared of 0.9346 and higher error values (MSE of 0.0583, RMSE of 0.2414), making it less effective for prediction. The **Artificial Neural Network (ANN)** significantly improved performance, achieving an R-squared of 0.9833 with MSE of 0.0149 and RMSE of 0.1220, capturing non-linear relationships more effectively. **Random Forest Regressor** outperformed all models with the highest R-squared value of 0.9980, MSE of 0.0018, and RMSE of 0.0421, indicating excellent prediction accuracy and precision. **XGBoost Regressor** showed comparable performance with a slightly higher R-squared value of 0.9986, MSE of 0.0013, and RMSE of 0.0359, making it a close contender to Random Forest, with marginally better test performance. Overall, both Random Forest and XGBoost provided the best predictive performance, with Random Forest slightly ahead in training accuracy and XGBoost excelling in test accuracy.



Conclusion

In conclusion, **Random Forest Regressor** and **XGBoost Regressor** outperformed all other algorithms, demonstrating superior prediction accuracy and lower error rates, with R-squared values close to 1, indicating they could explain nearly all of the variance in the target variable. **Artificial Neural Network (ANN)** also performed well, capturing non-linear relationships with a high R-squared value of 0.9833, but was outperformed by both Random Forest and XGBoost. The **linear models** (Linear Regression, Ridge Regression, and Lasso Regression) provided adequate results, with Linear and Ridge Regression yielding similar performance, while Lasso Regression, though useful for feature selection, slightly underperformed in prediction accuracy. Overall, for predicting calorie consumption, Random Forest Regressor and XGBoost Regressor were the most effective, with Random Forest slightly ahead due to its balance of performance and simplicity.

References

https://openai.com/chatgpt/

https://claude.ai/

https://www.tensorflow.org

https://scikit-learn.org

https://keras.io

https://matplotlib.org

https://seaborn.pydata.org

https://www.geeksforgeeks.org/calories-burnt-prediction-using-machine-learning