Predicting Calories Burned Using Machine Learning

1. Project type:
This project falls under the application-flavor category. It aims to develop a machine learning model to predict calories burned based on user characteristics and workout details. The project will use 3 different datasets, and 3 different machine learning models to analyze performance variations.

2. Problem statement:
Accurately estimating the number of calories burned during exercise is crucial for individuals tracking their fitness progress. Traditional methods, such as fitness trackers and metabolic equations, often provide generalized estimates that may not account for individual variations such as age, height, weight, and workout type. This project aims to leverage machine learning techniques to enhance calorie burn predictions based on available fitness data, optimizing the accuracy by experimenting with different models and datasets.

3. Project goal and motivation:
The goal of this project is to develop a machine learning model that estimates calorie expenditure more accurately than traditional methods by incorporating individual-specific attributes and workout details. By using multiple datasets and comparing different machine learning models, we aim to identify the best-performing approach.

4. Methodology and plan:

- Data Collection & Preprocessing**:**
    - Obtain three datasets containing features like age, height, weight, session duration, total calories burned, and workout type.
    - Clean the data by handling missing values, outliers, and standardize features.
- Model Selection & Training:
    - Implement three different machine learning models:
        - XGBoost – A machine learning model that is used for regression and classification tasks. This model was chosen because it works well with numerical and categorical features and it uses feature importance so we can see what features have the most impact on predictions.
        - Two types of Neural Networks
            - Feedforward Neural Network (MLP) – A type of neutral network that computes outputs for inputs. This neural network is chosen because it works well with both numerical and categorical features.
            - Recurrent Neural Network – A type of neural network that is designed remembering previous inputs which makes it good for sequential data. This neural network was chosen because it will work well with heart rate data that is in our datasets.

- Model Evaluation: Compare model performance using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared metrics**.**
- Hyperparameter Tuning**:**
    - Optimize learning rates, batch sizes, dropout rates, and regularization techniques for better performances.
    - Experiment with different optimizers.

Milestones & Schedule:
Phase 1: Data collection, cleaning, and preprocessing.
Phase 2: Exploratory data analysis and feature engineering.
Phase 3: Train and evaluate XGBoost and neural network models.
Phase 4: Compare performance across dataset sizes and optimize hyperparameters.
Phase 5: Integration and final testing.

5. Datasets:
Existing public datasets: We are using datasets from Kaggle containing different body features to see the relations between each other.
Each dataset contains key features such as age, gender, height, weight, session duration, workout type, heart rate (resting and average) and calories burned to help with calculating the prediction of calories burned.

- Dataset 1: https://www.kaggle.com/datasets/nadeemajeedch/fitness-tracker-dataset
    - Rows: 1800
    - Columns: 15
    - File Size: 0.13 MB
- Dataset 2: https://www.kaggle.com/datasets/adilshamim8/workout-and-fitness-tracker-data
    - Rows: 10000
    - Columns: 20
    - File Size: 0.93 MB
- Dataset 3: https://www.kaggle.com/datasets/valakhorasani/gym-members-exercise-dataset
    - Rows: 973
    - Columns: 15
    - File Size: 0.06 MB

6. Resources Needed:
Computational Resources**:** Python, PyTorch, Scikit-learn, Jupyter Notebook.
Dataset Sources: Public datasets from Kaggle.

7. Workload distribution:

TASK	WEI FAN WANG	DARSHAN NAIR	COLLABORATION
PHASE 1	Find and collect datasets from Kaggle.	Clean the dataset, handle missing value.	Perform initial data exploration.
PHASE 2	Visualize data distribution, and correlations.	Analyze trends, feature importance.	Discuss key findings for model design.
PHASE 3	Implement XGBoost model.	Implement two neural network models.	Compare model performance across datasets.
PHASE 4	Tune hyperparameter (learning rate, optimizers.)	Experiment with dataset size variations.	Compare performance and optimize models.
PHASE 5	Set up API for model deployment.	Create a simple UI.	Ensure model integration works.
FINAL REPORT & PRESENTATION	Write methodology, data preprocessing and EDA section.	Write model training, results, and conclusion sections.	Proofread, finalize, and prepare presentation.