**Status Report on Predicting Calories burned using machine learning**

* **Problem Statement**

The goal of this project is to develop a machine learning model to accurately predict calories burned based on various physiological and exercise-related factors. Current fitness tracking applications rely on general estimations that do not consider personalized data effectively. By leveraging machine learning, we aim to enhance the accuracy of calorie predictions.

* **Project Goal**

The primary objective is to build a predictive model that estimates calories burned based on input features such as age, gender, weight, height, heart rate metrics (Max\_BPM, Avg\_BPM, Resting\_BPM), session duration, workout type, body fat percentage, water intake, workout frequency, experience level, and BMI. The model will be evaluated for accuracy and optimized for performance to provide better estimations compared to standard fitness formulas.

* **Assumption and Methods**
* Assumption
* The data is clean and representative of real-world user.
* BMI are strong indicators of calorie burn.
* Different workout types impact calories burned differently.
* BMI and body fat percentage influence the rate at which calories are burned.
* Methods
* Exploratory Data Analysis to visualize feature relationships.
* Processing: Handling missing values, encoding categories features.
* Feature engineering: Normalization and one-hot encoding for categorical data.
* Model selection: XGBoost, FNN and XXX
* Hyperparameter tuning: Adjusting bench rate, batch size and regularization techniques.
* Performance evaluation using MAE, MSE and r2 score.
* **Software, Tools and Datasets**
* Programming languages and libraries:
* Python, Pandas, NumPy, Matplotlib, Seaborn (EDA & Preprocessing)
* Scikit Learn (AE, MSE, R2, Train Test Split)
* TensorFlow/Pytorch (Deep Learning Models)
* Dataset: Fitness data including heart rates, workout type and calories burned.
* **Experimental Plan**
* Step 1 - Data Analysis and Preprocessing:
* Perform EDA to understand feature distribution and correlations.
* Clean data by removing inconsistencies and handling missing values.
* Encode categorical variables.
* Normalize numerical features.
* Step 2 - Model training and Benchmarking:
* Train and compare XGBoost, FNN and XXX
* Tune hyperparameters (learning rates, dropout layers, batch size)
* Evaluate performance using MAE, MSE and R2 score.
* Step 3 - Optimization and fine-tuning:
* Identify and address model biases.
* Use feature importance analysis to refine model input.
* Implement ensemble method if needed.
* Step 4 - Testing and Deployment:
* Validate the model with test data.
* Deploy in a simple application or API for real-time predictions.
* **Current Status and Partial Results**
* EDA completed: Identified correlations.
* Data preprocessing: Handling missing values and encoding categorical data (one-hot encoding applied).
* Feature Engineering: Normalization of numeric features (currently working on it).
* Initial Bench Marking:
* XGBoost
* FNN
* XXX
* **Plan for the remaining month**
* Week 1: Finalize hyperparameter tuning for all models.
* Week 2: Optimize FNN for better generalization.
* Week 3: Conduct cross validation and compare models thoroughly.
* Week 4: Finalize results, create a presentation/report, and prepare for the potential deployment.