**Technical Report: Calories Prediction System**

**Overview:**

The Calories Prediction System is a machine learning-based solution designed to predict the number of calories burned based on various input factors, such as gender, age, height, weight, duration of exercise, heart rate, and body temperature. The system compares the performance of multiple regression models and selects the best one to provide accurate predictions.

**Project Structure:**

* **Data:**
  + The input data is stored in exercise.csv, which contains columns like Gender, Age, Height, Weight, Duration, Heart\_Rate, Body\_Temp, and Calories.
  + The target variable is Calories, which represents the number of calories burned.
* **Modules and Libraries:**
  + numpy and pandas for data manipulation.
  + seaborn and matplotlib for data visualization.
  + sklearn for model training and evaluation (Linear Regression, Random Forest, Decision Tree).
  + xgboost for additional model training.
  + pickle for model serialization.

**Program Flow:**

1. **Data Loading:** The dataset is loaded using pandas.read\_csv(), and the first few rows are inspected using exercise.head().
2. **Exploratory Data Analysis (EDA):**
   * Scatter plots, distribution plots, and count plots are generated to understand the relationship between variables.
   * A correlation matrix is computed and visualized using a heatmap to check for correlations between numeric features.
3. **Data Preprocessing:**
   * Gender is mapped to numeric values (0 for male, 1 for female).
   * The dataset is split into features (X) and the target variable (y).
   * The data is then divided into training and testing sets using train\_test\_split().
4. **Model Training:**
   * Four models are trained: Linear Regression, RandomForestRegressor, DecisionTreeRegressor, and XGBRFRegressor.
   * The Mean Squared Error (MSE) and R2 scores are calculated for each model to evaluate performance.
5. **Model Selection:**
   * Based on the evaluation metrics, RandomForestRegressor is selected as the best model.
6. **Predictive System:**
   * A pred() function is created, allowing the user to input values for the features and get a predicted calorie count using the trained Random Forest model.
7. **Model Serialization:**
   * The trained Random Forest model is saved using pickle, so it can be used later without retraining.
8. **Visualization:**
   * A bar chart compares the R2 scores of the different algorithms for a visual understanding of their performance.

**Key Files and Functions in the Project:**

* **exercise.csv:** The dataset used for training and testing the models.
* **pred() function:** Takes input features and returns a predicted calorie count.
* **model.pkl:** A serialized version of the trained RandomForestRegressor.
* **Visualization plots:** Used to provide insights into the data and model performance.

**Instructions to Run the Code:**

1. **Install necessary libraries** by running:

pip install numpy pandas seaborn matplotlib scikit-learn xgboost

1. **Run the script** by executing it in a Python environment:

python calories\_prediction.py

1. **To make a prediction:** Use the pred() function with the required input values:

result = pred(1, 36, 167.0, 64.0, 28.0, 108.0, 40.0)

print("Predicted Calories:", result)

1. **To load the saved model** and use it for predictions in a separate script:

python

import pickle

model = pickle.load(open('model.pkl', 'rb'))

**Sample Output:** Upon running the system, the following outputs are expected:

**Model Evaluation:**

lr mse: 130.087 r2 score: 0.9669

rfr mse: 6.933 r2 score: 0.9982

dtr mse: 27.650 r2 score: 0.9930

xg mse: 58.364 r2 score: 0.9851

**Sample Prediction:**

Predicted Calories: 266.75

**Error Handling:**

* **File Not Found Error:**
  + If exercise.csv is not found, a FileNotFoundError will occur. Ensure that the file is in the correct directory.
* **Input Value Error:**
  + If incorrect data types are passed into the pred() function (e.g., strings instead of numbers), the system will raise a ValueError. Ensure all inputs are numeric.
* **Model Not Trained:**
  + If the model is not trained or the model file (model.pkl) is missing, the system will raise an error. Ensure the model is trained before saving it.

**Conclusion:**

This project successfully implements a multi-model comparison approach to predict calories burned based on personal and exercise-related features. The Random Forest Regressor provides the best performance with an R2 score of 0.9982. The model can be serialized and reused, making it highly efficient for real-time predictions in health-related applications.