Step Tracking System: Data Processing, Model Development & Deployment

1. Data Collection & Preprocessing

Data Source

We have used a Kaggle dataset containing step tracking data, including features such as total steps, active minutes, distance covered, and calorie burn. The dataset was preprocessed to remove noise and ensure quality.

Data Cleaning

- **Handled missing values using interpolation and mean imputation.**
- **Removed outliers using Z-score analysis.**
- **Normalized step count and distance values using Min-Max scaling.**

Feature Engineering

- **Extracted step frequency and rest periods to enhance model performance.**
- **Created activity duration metrics to better classify activity patterns.**

2. Model Development

Model Selection

We experimented with multiple models, including Decision Tree, Random Forest, XGBoost, LSTM, and CNN. Ultimately, LSTM was chosen for its ability to capture temporal dependencies in step data.

Hyperparameter Tuning

- **Used GridSearchCV for hyperparameter tuning in traditional models.**
- **For LSTM, optimized batch size, number of layers, and dropout rates to prevent overfitting.**

Performance Metrics

- **Mean Absolute Error (MAE) and Mean Squared Error (MSE) were used to evaluate regression accuracy.**
- **Compared models based on performance, and LSTM showed the best results for step prediction.**

3. Integration with Firmware & App

Deployment on Marshee Smart Tracker

- **The trained LSTM model is deployed using TensorFlow Lite for efficient inference on embedded devices.**
- **Real-time inference is processed either on-device or via a cloud-based API when higher accuracy is needed.**

Streamlit UI Development

We developed a Streamlit-based UI to visualize real-time step data, allowing users to:

- **Upload step tracking data.**
- **View insights like daily step trends and activity levels.**
- **Get predictions and fitness recommendations.**
- **Monitor progress with an interactive dashboard.**

Embedded System Optimization

- **Quantized the model to reduce memory usage.**
- **Used edge computing to reduce cloud dependency.**

4. Deployment Strategy

OTA Updates

- **The model is updated over-the-air (OTA) to ensure real-time improvements.**
- **A CI/CD pipeline is established for seamless firmware upgrades.**

Cloud-Based Architecture

- **A Flask API serves real-time predictions for the mobile app.**
- **Firebase is used for storing step data logs and analytics.**

Periodic Retraining

- **The model retrains using new data every month.**
- **User feedback helps improve predictions over time.**

5. Additional Considerations

Challenges & Solutions

- **Real-time Processing: Optimized inference using TFLite.**
- **Limited Compute Power: Used pruning & quantization to reduce model size.**

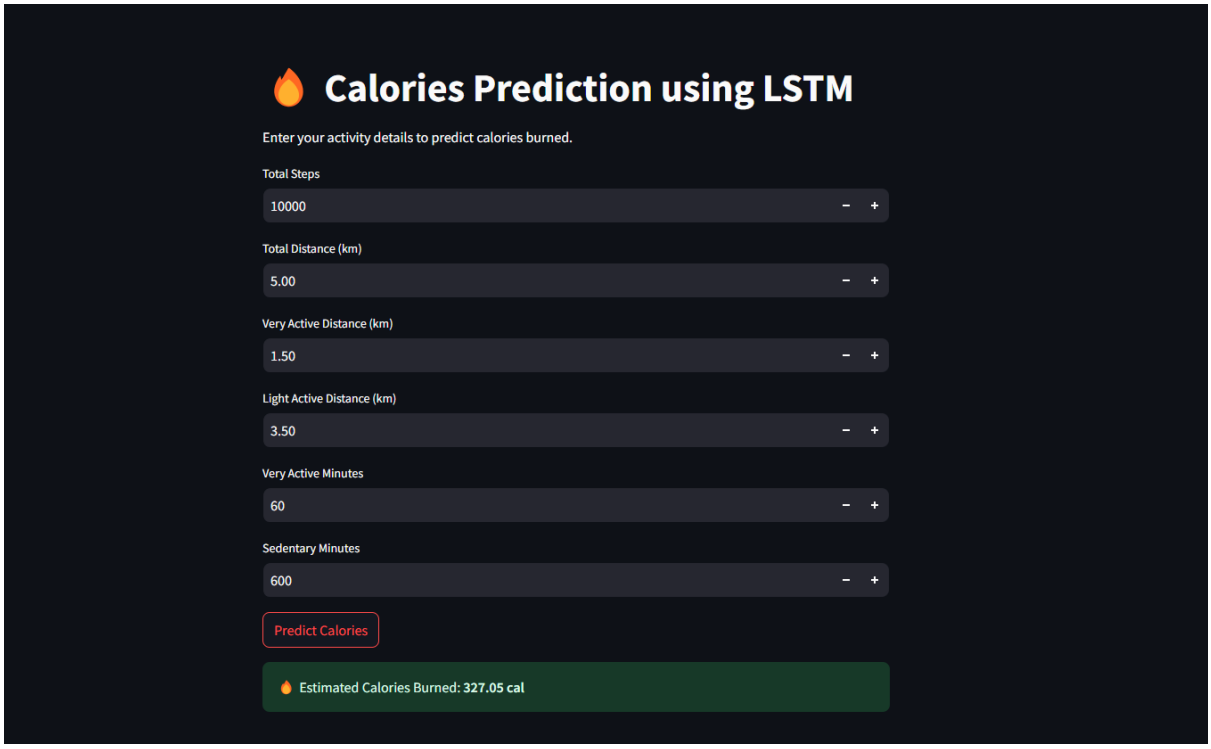
Security & Privacy

- **Implemented AES encryption for sensitive step data.**
- **Adhered to GDPR-compliant data handling policies.**

AI-Driven Features

- **Step anomaly detection to identify irregular activity.**
- **Personalized fitness recommendations based on trends.**

Streamlit UI & Demo


Calories Prediction using LSTM
Enter your activity details to predict calories burned.
Total Steps
10000
Total Distance (km)
5.00
Very Active Distance (km)
1.50
Light Active Distance (km)
3.50
Very Active Minutes
60
Sedentary Minutes
600
Predict Calories
Estimated Calories Burned: 327.05 cal