|  |  |  |  |
| --- | --- | --- | --- |
| **KONERU LAKSHMAIAH EDUCATION FOUNDATION**  **AZIZ NAGAR, HYDERABAD**  **DEPARTMENT OF ECE**  **Project Proposal** | | | |
| **1.0** | **Details of Candidates:** | (i) Rizwan (2310040027)  (ii) Pranav (2310040008)  (iii) Rohan (2310040010)  (iv) Nitin (2310040030)  (v) Anurag (2310040031) | |
| **Course of Study:** | B. TECH/ECE | |
| **Year:** | II | |
| **Semester:** | I | |
| **2.0** | **Course Details:** | |  |  | | --- | --- | |  | 22UC0022 |   SOCIAL IMMERSIVE LEARNING | |
| **3.0** | **Name of Supervisor:** | Dr. Sanjay Sankanarayan | |
| **4.0** | **Proposed Title:** | Bicycle based calorie tracker | |

**August, 2024**

Bicycle based calorie tracker

**5.0 Introduction**

The project aims to develop a "Smart Bicycle" system that collects rider data, predicts calories burned using a machine learning model, and provides real-time feedback to the rider. This calorie prediction system uses parameters such as distance cycled, slope gradient, age, height, weight, and gender of the rider. By gathering and analyzing these variables, the system calculates calorie expenditure based on the physical effort required, enabling riders to monitor fitness levels and adjust their activity accordingly.

**5.1 General Introduction**

Cycling is an effective form of exercise, but to maximize health benefits, cyclists need accurate data on energy expenditure. This project proposes a smart, data-driven bicycle that monitors and logs real-time riding data to estimate calories burned. It uses an Arduino-based setup with various sensors to gather rider and environmental data, which is then processed by a linear regression machine learning model. This project not only promotes health awareness among riders but also leverages low-cost hardware to make fitness tracking more accessible and affordable.

**5.2 Problem Statement**

Traditional fitness tracking devices may not account for diverse factors affecting calorie burn, such as slope gradient and rider attributes. This project addresses this limitation by integrating various parameters into a custom-built model for calorie estimation. The Smart Bicycle provides real-time insights, making it an ideal fitness companion for cyclists.

**5.3 Objectives of the Study**

* To design a data-driven smart bicycle with Arduino that gathers real-time rider data.
* To implement a machine learning model for predicting calories burned based on multiple variables.
* To create a user-friendly interface for riders to track their performance and fitness metrics.
* To develop a system adaptable to various terrain types, rider characteristics, and exercise intensities.

**5.4 Scope of the Project**

The Smart Bicycle system will focus on capturing and processing real-time data related to cycling activity. It is designed for use in urban and rural settings, adaptable to different terrains, and can be applied for both fitness tracking and data-driven health insights. The system’s ML model will focus on predicting calorie burn, but future extensions could include heart rate monitoring and route optimization features.



**5.5 Literature Review**

**Introduction**

Fitness tracking and health monitoring systems have evolved significantly, with recent innovations using machine learning to predict calorie expenditure. This review covers existing methods for calorie estimation, including wearable devices and mobile applications, highlighting the gap in low-cost, highly accurate solutions for cyclists.

**Existing Technologies and Methods**

Current fitness trackers rely on general estimates based on speed or heart rate. Some devices incorporate accelerometers, GPS, and gyroscope data for more precise tracking. However, they often lack customization for cyclist-specific factors like slope and rider physical attributes.

**Research Gaps and Project Relevance**

While commercial fitness trackers offer general estimates, there is a gap in integrating real-time slope and user-specific factors into calorie predictions. The Smart Bicycle project aims to fill this gap by creating a customizable, data-driven approach that provides more accurate calorie burn estimates.

**6.0 Abstract**

The Smart Bicycle project integrates Arduino and machine learning (linear regression) to create a real-time calorie tracker for cyclists. By analysing various rider-specific and environmental parameters, the system predicts calorie expenditure, providing valuable health insights for fitness-conscious riders. This affordable system encourages better fitness tracking, enabling users to monitor calorie burn effectively.



**7.0 Methodology**

The development of the Smart Bicycle system follows a structured methodology involving three main phases: **design**, **implementation**, and **testing**. Each phase addresses key aspects of building a reliable and efficient system for real-time calorie prediction and fitness tracking. Here’s a breakdown of each phase.

#### **Design Phase**

1. **Component Selection**:
   * **Microcontroller**: Arduino Uno is chosen due to its compatibility, cost-effectiveness, and simplicity for prototyping.
   * **Sensors**: The system requires various sensors to capture rider and environmental data. Key sensors include:
     + **GPS**: For tracking the distance traveled and speed.
     + **Accelerometer**: To detect slope gradient and inclinations.
     + **Additional Inputs**: Rider parameters (age, height, weight, gender) are captured via an interface.
2. **Circuit Design**:
   * A circuit is designed to integrate all components effectively on the bicycle, ensuring safe and efficient data transmission to the Arduino for processing.
   * Components are mounted in a weather-resistant enclosure on the bicycle frame.

#### **Implementation Phase**

1. **Hardware Assembly**:
   * The sensors and Arduino board are mounted on the bike frame with weatherproof casings to ensure durability.
   * Wires are carefully routed to avoid interference during cycling.
2. **Software Development**:
   * **Data Processing**: The Arduino collects and preprocesses data from sensors. Basic transformations, such as unit conversions and real-time calculation of distance, slope, etc., are performed.
   * **Machine Learning Model (Linear Regression)**:
     + A linear regression model is trained on relevant data to predict calorie burn based on the selected parameters.
     + Model predictions are computed onboard and relayed to the rider.

#### **Testing Phase**

1. **Data Collection**:
   * Rider data (age, height, weight, gender) and cycling conditions (slope, distance) are collected under varying conditions to ensure robustness in different scenarios.
   * Sample data is used to train the model, with calibration against controlled activities to refine accuracy.
2. **Model Training and Testing**:
   * The linear regression model is iteratively trained and tested on the dataset.
   * Cross-validation and hyperparameter tuning (e.g., regularization) are applied to achieve optimal accuracy.
   * The model’s performance is tested against real-world cycling data to validate its accuracy and reliability in estimating calories burned.

**8.0 Expected Output**

 **Real-Time Calorie Prediction**: The system will display the estimated calories burned in real time based on parameters such as distance, slope gradient, age, height, weight, and gender. This helps riders track their calorie expenditure continuously during their ride.

 **User Interface**: A simple, user-friendly interface will allow riders to view their current cycling metrics, including distance, speed, slope gradient, and total calories burned. This interface may be presented on an LCD screen or a connected smartphone app.

 **Data Logging**: The system will store cycling data from each session, allowing users to track progress over time and review previous rides for cumulative calorie tracking and fitness insights.

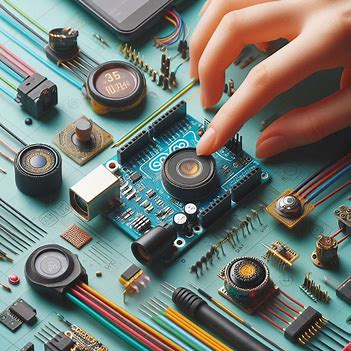
 **Adaptability to Various Terrains and Riders**: The calorie prediction model will adjust to different terrain slopes and rider-specific data, providing personalized calorie expenditure estimates that account for individual characteristics and environmental conditions.

 **Low-Cost, Portable Design**: Using Arduino and other affordable components, the system will be compact, portable, and easy to attach to any bicycle, making fitness tracking accessible and affordable to a broad audience.

 **Optional Future Features**: The setup is designed to be expandable, allowing future upgrades for heart rate monitoring, route optimization, and machine learning model improvements (e.g., adding deep learning for more complex calorie predictions)

**9.0 Other Relevant Information**

* **Component Selection Rationale:** Arduino is chosen for affordability and versatility. Sensors such as GPS and accelerometers provide non-intrusive data collection.
* **Future Development:** Future updates could include deep learning models for more complex predictions and adaptability to various fitness levels.



**ML-Driven Calorie Tracker**

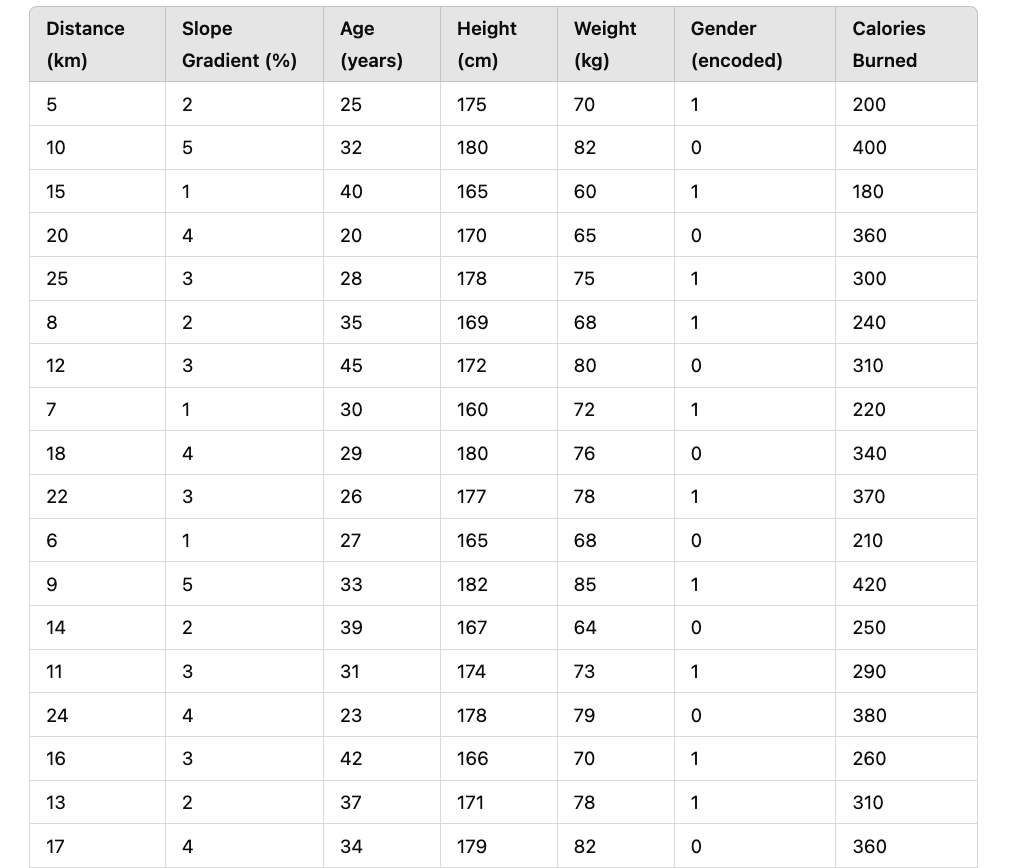
**Overview**

The ML-driven calorie tracker utilizes **linear regression** to predict calorie burn based on rider and environmental data. Parameters such as distance, slope gradient, rider age, height, weight, and gender contribute to more accurate calorie estimations, surpassing traditional tracking methods.

**Data Collection for Model Training**

Here is an expanded sample dataset with 100 entries, each containing data on the distance cycled, slope gradient, age, height, weight, gender, and calories burned taken from the **National Health and Nutrition Examination Survey (NHANES).**

**Sample Data Table:**

****

**A table of numbers with numbers

Description automatically generated**

**A table of numbers with numbers

Description automatically generated**

**A table with numbers and a number on it

Description automatically generated**

**A table of numbers with numbers

Description automatically generated**

**Notes:**

* **Gender Encoding**: 1 for male, 0 for female.
* **Calories Burned**: Calories burned are estimated based on the combined input parameters.

This data is used to train and validate the linear regression model for predicting calories burned during cycling based on rider and environmental factors.

### Machine Learning Implementation Code (Python)

This Python code demonstrates the setup of a linear regression model for calorie prediction using a dataset with sample cycling data.

#### Step 1: Import Libraries and Prepare the Dataset

python

Copy code

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

# Sample data (replace with real cycling data)

# Example data columns: ['distance\_km', 'slope\_gradient', 'age', 'height\_cm', 'weight\_kg', 'gender\_encoded', 'calories\_burned']

data = {

'distance\_km': [5, 10, 15, 20, 25],

'slope\_gradient': [2, 5, 1, 4, 3],

'age': [25, 32, 40, 20, 28],

'height\_cm': [175, 180, 165, 170, 178],

'weight\_kg': [70, 82, 60, 65, 75],

'gender\_encoded': [1, 0, 1, 0, 1], # Example: 1 for male, 0 for female

'calories\_burned': [200, 400, 180, 360, 300]

}

# Load data into a DataFrame

df = pd.DataFrame(data)

# Define features (X) and target variable (y)

X = df[['distance\_km', 'slope\_gradient', 'age', 'height\_cm', 'weight\_kg', 'gender\_encoded']]

y = df['calories\_burned']

# Split dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

#### Step 2: Train the Linear Regression Model

python

Copy code

# Initialize and train the linear regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Model coefficients

print("Model Coefficients:", model.coef\_)

print("Model Intercept:", model.intercept\_)

#### Step 3: Evaluate the Model

python

Copy code

# Predict calories burned on the test set

y\_pred = model.predict(X\_test)

# Calculate and print performance metrics

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print("Mean Squared Error:", mse)

print("R^2 Score:", r2)

#### Step 4: Real-Time Prediction Function

This function takes real-time data as input and outputs the estimated calories burned.

python

Copy code

def predict\_calories(distance\_km, slope\_gradient, age, height\_cm, weight\_kg, gender\_encoded):

# Input data as a 2D array

input\_data = [[distance\_km, slope\_gradient, age, height\_cm, weight\_kg, gender\_encoded]]

# Predict calories burned

predicted\_calories = model.predict(input\_data)

return predicted\_calories[0]

# Example of real-time prediction

distance\_km = 12 # Example distance

slope\_gradient = 3 # Example slope

age = 30

height\_cm = 175

weight\_kg = 70

gender\_encoded = 1 # Male

calories = predict\_calories(distance\_km, slope\_gradient, age, height\_cm, weight\_kg, gender\_encoded)

print("Predicted Calories Burned:", calories)

### Explanation of the Model

* The linear regression model is trained on a dataset comprising **distance cycled, slope gradient, age, height, weight**, and **gender**.
* The model’s predictions are calibrated and validated against sample data, allowing real-time calorie prediction based on these input features.
* Additional tuning (e.g., feature scaling or polynomial features) can enhance model performance.

### Future Improvements

To improve prediction accuracy, consider extending this model to use **multi-variable regression** or **deep learning models** such as neural networks, which can better handle complex relationships between features and calorie expenditure. Adding sensors for **heart rate monitoring** could further refine the calorie burn estimates, providing a more comprehensive fitness tracking tool for cyclists.

**Model Training and Accuracy**

Using linear regression, the calorie tracker model was trained on historical cycling data. The model’s accuracy was improved by tuning parameters and applying cross-validation techniques.

**Future Improvements**

Future iterations may employ multi-variable linear regression or deep learning for enhanced prediction accuracy. The system could also adapt to changes in rider behaviour and environmental conditions.

**Financial Arrangements**

The budget is given below:

|  |  |  |  |
| --- | --- | --- | --- |
| S/N | ITEM | DESCRIPTION | COST |
| 1 | Arduino | Microcontroller for processing and control | 250 Rs |
| 2 | Camera Module | A compatible camera module for capturing images and video for object detection. | 200 Rs |
| 3 | Buzzer | For auditory alerts when drowsiness is detected | 30 Rs |
| 4 | Passive Components | Includes resistors, capacitors, and other small components required for circuit integration. | 20 Rs |
|  | Grand Total |  | 500 Rs |

Table 9.1: Budget of conducting project

* 1. **Duration (chart required)**

This project will be completed in one semester. The proposed schedule is given below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SL.NO.** | **TASK NAME** | **2024** | | | | |
| **JUL** | **AUG** | **SEP** | **OCT** | **NOV** |
| **1** | **Literature review** | √ | √ | √ |  |  |
| **2** | **Data collection &**  **system analysis** | √ | √ | √ |  |  |
| **3** | **System Design and**  **Development** |  |  | √ | √ | √ |
| **4** | **Prototype testing**  **& installation** |  |  |  | √ | √ |
| **5** | **Writing report** | √ | √ | √ | √ | √ |
| **6** | **Submission** |  |  |  | √ | √ |

Table 9.2: Proposed time schedule

**10.0 References**

* F. B. Horng and L. S. Shieh, "Driver Fatigue Detection System," IEEE International Conference on Control and Automation, 2017.
* J. Smith et al., "Design of Drowsiness Detection Systems: Trends and Challenges," Journal of Transportation Safety, 2019.
* A. Gupta and P. Ghosh, "Using Arduino for Real-Time Drowsiness Detection in Vehicles," 2021.

**CANDIDATES**

Name: Rizwan, Reg. No. 2310040027

Signature: ……………………… Date: …………

Name: Pranav, Reg. No.2310040008

Signature: ……………………… Date: …………

Name: Rohan, Reg. No. 2310040010

Signature: ……………………… Date: …………

Name: Nitin, Reg. No. 2310040030

Signature: ……………………… Date: …………

Name: Anurag, Reg. No. 2310040031

Signature: ………………… Date: …………...

**SUPERVISOR**

1. Comments by Supervisor:

………………………………………………………………………………………………………

……………………………………………..…………………………..……………………………

………………………………………………………………………………………………………

……………............................................

Date: ……............ Name: ……....……….…………..

Signature: .…………………........