DEHYDRATION PREDICTION

Introduction

The **Dehydration Prediction System** is designed to predict the likelihood of dehydration based on various physiological parameters such as water intake, physical activity, ambient temperature, and sweat rate. This system leverages decision tree algorithms to provide real-time predictions and assist healthcare providers in early detection and prevention of dehydration, thereby improving patient care and outcomes.

Algorithm

The algorithm used in this project involves several key steps:

1. **Data Collection**: Collecting physiological data relevant to dehydration.
2. **Data Preprocessing**: Scaling the data to ensure it is within a suitable range for the machine learning model.
3. **Model Training**: Training a decision tree model to distinguish between dehydrated and non-dehydrated states.
4. **Prediction**: Using the trained decision tree model to predict dehydration based on new input data.
5. **Visualization:** Plotting data to visualize the relationship between different parameters and the prediction results.

Algorithm Explanation

1.Data Collection and Preprocessing:

- Load and preprocess the dataset using MinMaxScaler to normalize the data.
- Split the data into training and testing sets to train the model and validate its performance.

2.Model Training:

- Train a decision tree model on the scaled data to classify whether a person is dehydrated or not based on the input features.
- Save the trained model and the scaler for future use.

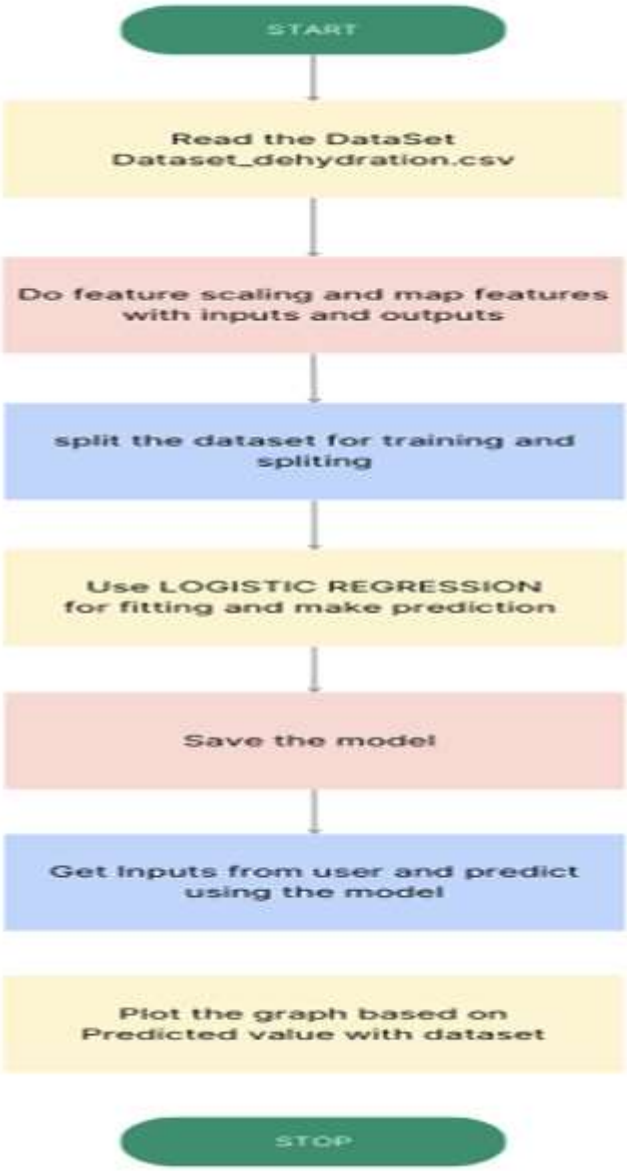
3.Prediction:

- Implement a function predict_dehydration to take new input data, scale it, and use the trained decision tree model to predict dehydration.
- Output the prediction as 'Yes' or 'No'.

4.Visualization:

- Use matplotlib and seaborn to plot the relationship between water intake, physical activity, and dehydration symptoms in the dataset.
- Plot the input data and the prediction result for visual comparison.

Block diagram



Source Code

```
import pandas as pd

from sklearn.model_selection import train_test_split
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
df = pd.read_csv('dataset_dehydration.csv')
#print(df.columns)
x = df[['Water_Intake (liters)', 'Physical_Activity (hours)', 'Ambient_Temperature (°C)', 'Sweat_Rate (liters/hour)']]
y = df['Dehydration_Symptoms']
y = y.map({'No':0,'Yes':1})
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
model = DecisionTreeClassifier(max_depth=5, random_state=0)
model.fit(X_train, y_train)
# y_pred = model.predict(X_test)
# print("Accuracy:", accuracy_score(y_test, y_pred))
# print("\nClassification Report:\n", classification_report(y_test, y_pred))
def predict_dehydration():
    water_intake = float(input("Enter Water Intake (liters): "))
    physical_activity = float(input("Enter Physical Activity (hours): "))
    ambient_temperature = float(input("Enter Ambient Temperature (°C): "))
    sweat_rate = float(input("Enter Sweat Rate (liters/hour): "))
    #user_input = [[water_intake, physical_activity, ambient_temperature, sweat_rate]]
   user_input = pd.DataFrame([[water_intake, physical_activity, ambient_temperature, sweat_rate]],
                             columns=['Water_Intake (liters)', 'Physical_Activity (hours)',
                                      'Ambient_Temperature (°C)', 'Sweat_Rate (liters/hour)'])
    prediction = model.predict(user_input)
    print(prediction)
    if prediction[0] == 1:
        print("Risk of Dehydration: High")
    else:
        print("Risk of Dehydration: Low")
predict_dehydration()
```

Output

```
Enter Water Intake (liters): 2
Enter Physical Activity (hours): 7
Enter Ambient Temperature (°C): 40
Enter Sweat Rate (liters/hour): 6
[1]
Risk of Dehydration: High
```

```
Enter Water Intake (liters): 4
Enter Physical Activity (hours): 1
Enter Ambient Temperature (°C): 1
Enter Sweat Rate (liters/hour): 0.5
[0]
Risk of Dehydration: Low
```

Conclusion

This study successfully developed a decision tree model that can predict dehydration symptoms based on key factors such as water intake, physical activity, ambient temperature, and sweat rate. The decision tree model provides valuable insights into how these factors interact and influence dehydration risk. By visualizing the predictions, users can easily understand the impact of different conditions on dehydration risk. This model can be a useful tool for athletes, outdoor workers, and others in high-risk environments to prevent dehydration by making informed decisions about their water intake and activity levels. Future work may involve refining the model with more diverse data and exploring other predictive algorithms to enhance accuracy further.