Project 1: Extracting Time Series Properties of Glucose Levels in Artificial Pancreas Introduction

Purpose

In this project, you will extract several performance metrics of an Artificial Pancreas system from sensor data.

Objectives

Learners will be able to:

- Extract feature data from a data set.
- Synchronize data from two sensors.
- Compute and report overall statistical measures from data.

Technology Requirements

- Python 3.6 to 3.8 (do not use 3.9).
- scikit-learn==0.21.2
- pandas==0.25.1
- Python pickle

Project Description

In this project, we are considering the Artificial Pancreas medical control system, specifically the Medtronic 670G system. The Medtronic system consists of a continuous glucose monitor (CGM) and the Guardian Sensor (12), which is used to collect blood glucose measurements every 5 minutes. The sensor is single-use and can be used continuously for 7 days after which it has to be replaced. The replacement procedures include a recalibration process that requires the user to obtain blood glucose measurements using a Contour NextLink 2.4 glucosemeter ®.

Note that this process also requires manual intervention. The Guardian Link the CGM sensor and sends the data to the MiniMed 670G® insulin pump. T utilizes the Smart Guard Technology that modulates the insulin delivery bas The SmartGuard Technology uses a Proportional, Integrative, and Derivativ small bursts of insulin also called Micro bolus to be delivered to the user. Du uses a BolusWizard to compute the amount of food bolus required to maint levels. The user manually estimates the amount of carbohydrate intake and Wizard.

The Bolus Wizard is pre-configured with the correction factor, body weight, sensitivity of the subject, and it calculates the bolus insulin to be delivered. program the MiniMed 670G infusion pump to deliver that amount. In additio MiniMed 670G insulin pump can also provide a correction bolus. The correc provided only if the CGM reading is above a threshold (typically 120 mg/dL) amount with respect to the difference of the CGM reading and the threshold

The SmartGuard technology has two methods of suspending insulin deliver where the insulin delivery is stopped when the CGM reading is less than a c suspend on predicted low, where the insulin delivery is stopped when the C predicted to be less than a certain threshold. Apart from these options, insu suspended manually by the user or can be suspended when the insulin res

Directions

Dataset:

You will be given two datasets:

1. From the Continuous Glucose Sensor (CGMData.csv) and
2. from the insulin pump (InsulinData.csv)

The output of the CGM sensor consists of three columns:

1. **Data time stamp (Columns B and C combined)**,
2. **the 5 minute filtered CGM reading in mg/dL**, (Column AE) and
3. the ISIG value which is the raw sensor output every 5 mins.

The output of the pump has the following information:

1. **Data time stamp**,
2. Basal setting,
3. Micro bolus every 5 mins,
4. Meal intake amount in terms of grams of carbohydrate,

5. Meal bolus,
6. correction bolus,
7. correction factor,
8. CGM calibration or insulin reservoir-related alarms, and
9. **auto mode exit events and unique codes representing reasons (**

The bold items are the columns that you will be using in this assignment.

Metrics to be extracted:

1. Percentage time in hyperglycemia (CGM > 180 mg/dL),
2. percentage of time in hyperglycemia critical (CGM > 250 mg/dL),
3. percentage time in range (CGM >= 70 mg/dL and CGM <= 180 mg/c
4. percentage time in range secondary (CGM >= 70 mg/dL and CGM <
5. percentage time in hypoglycemia level 1 (CGM < 70 mg/dL), and
6. percentage time in hypoglycemia level 2 (CGM < 54 mg/dL).

Each of the above-mentioned metrics are extracted in three different time in
to midnight), overnight (midnight to 6 am), and whole day (12 am to 12 am).

Percentage is with respect to the total number of CGM data that should be a
Assume that the total number of CGM data that should be available is 288.
such that the number of data available is less than 288, but still consider the
respect to 288.

You have to extract these metrics for each day and then report the mean va
over all days. Hence there are 18 metrics to be extracted.

The metrics will be computed for two cases:

- Case A: Manual mode
- Case B: Auto mode

Analysis Procedure:

The data is in reverse order of time. This means that the first row is the end
whereas the last row is the beginning of the data collection. The data starts
Manual mode continues until you get a message “AUTO MODE ACTIVE PL
column “Q” of the InsulinData.csv. From then onwards Auto mode starts. *Yo*
“AUTO MODE ACTIVE PLGM OFF” *in column “Q” but only use the earliest*
when you switch to auto mode. There is no switching back to manual mode
determine the time stamp when Auto mode starts. **Remember that the tim**

data is not the same as the timestamp of the insulin pump data becaus different devices which operate asynchronously.

Once you determine the start of Auto Mode from InsulinData.csv, you l timestamp in CGMData.csv where Auto mode starts. This can be done for the time stamp that is nearest to (and later than) the Auto mode sta obtained from InsulinData.csv.

For each user, CGM data is first parsed and divided into segments, where e corresponds to a day worth of data. One day is considered to start at 12 am If there is no CGM data loss, then there should be 288 samples in each seg a whole is used to compute the metrics for the whole day time period. Each divided into two sub-segments: daytime sub-segment and overnight subseg subsegment, the CGM series is investigated to count the number of sample ranges specified in the metrics. To compute the percentage with respect to number of samples in the specified range is divided by 288.

Note that here you have to tackle the “missing data problem”, so a particula data points. In the data files, those are represented as NaN. You need to de tackle the missing data problem. Popular strategies include deletion of the e interpolation.

Write a Python script that accepts two CSV files: CGMData.csv and InsulinD analysis procedure and outputs the metrics discussed in the metrics sectior using the format described in Results.csv.

Submission Directions for Project Deliv

Submit your Python script and Results.csv file on Canvas.

Deliverables:

- Code (‘main.py.’ code file)
- Results.csv
  - Do NOT include headers
  - Your Results.csv should only contain numbers and it should be a 2 X
- Requirements file (detailing how the requirements were fulfilled). See att file template for an example. Save the Requirements file as a "Requirem

Submission Guidelines:

- Please submit a zipped file containing 1, 2, and 3 deliverables as "**yourfirstname_lastname_Project1.zip**".
  - Do not create an additional folder; just zip the files directly.
- **The submission space is located at the bottom of module Assignment: Project 1: Extracting Time Series Properties Levels in Artificial Pancreas Submission".**

Rubric:

Assignment 1 - Rubric			
Criteria	Ratings		
Successful execution of code with csv file generated	6 pts Full Marks	0 pts No Execution	6
Percentage time in hyperglycemia (CGM > 180 mg/dL) for auto and manual mode	4 pts Full Marks	0 pts No Marks	4
Percentage of time in hyperglycemia critical (CGM > 250 mg/dL) for auto and manual mode	4 pts Full Marks	0 pts No Marks	4
Percentage time in range (CGM >= 70 mg/dL and CGM <= 180 mg/dL) for auto and manual mode	4 pts Full Marks	0 pts No Marks	4
Percentage time in range secondary (CGM >= 70 mg/dL and CGM <= 150 mg/dL) for auto and manual mode	4 pts Full Marks	0 pts No Marks	4
Percentage time in hypoglycemia level 1 (CGM < 70 mg/dL) for auto and manual mode	4 pts Full Marks	0 pts No Marks	4
Percentage time in hypoglycemia level 2 (CGM < 54 mg/dL) for auto and manual mode	4 pts Full Marks	0 pts No Marks	4

Template_Requirements-1.docx (https://canvas.asu.edu/courses/140907/fil

Project-1-Files-1.zip (https://canvas.asu.edu/courses/140907/files/6072898
(https://canvas.asu.edu/courses/140907/files/60728989/download?download_f