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
Medtronic system consists of a continuous glucose monitor (CGM) and the Guardian Sensor (12), which is used
glucose measurements every 5 minutes. The sensor is single-use and can be used continuously for 7 days after
replaced. The replacement procedures include a recalibration process that requires the user to obtain blood gluc
using a Contour NextLink 2.4 glucosemeter ®.

Note that this process also requires manual intervention. The Guardian Link Transmitter®, powers the CGM sens
to the MiniMed 670G® insulin pump. The insulin pump utilizes the Smart Guard Technology that modulates the in
on the CGM data. The SmartGuard Technology uses a Proportional, Integrative, and Derivative controller to deriv
insulin also called Micro bolus to be delivered to the user. During meals, the user uses a BolusWizard to compute
bolus required to maintain blood glucose levels. The user manually estimates the amount of carbohydrate intake
Bolus Wizard.

The Bolus Wizard is pre-configured with the correction factor, body weight, and average insulin sensitivity of the
calculates the bolus insulin to be delivered. The user can then program the MiniMed 670G infusion pump to deliv
addition to the bolus, the MiniMed 670G insulin pump can also provide a correction bolus. The correction bolus a
if the CGM reading is above a threshold (typically 120 mg/dL) and is a proportional amount with respect to the dif
reading and the threshold.

The SmartGuard technology has two methods of suspending insulin delivery: a) Suspend on low, where the insu
when the CGM reading is less than a certain threshold, or b) suspend on predicted low, where the insulin delivery
CGM reading is predicted to be less than a certain threshold. Apart from these options, insulin delivery can also t
manually by the user or can be suspended when the insulin reservoir is running low.

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 (Column Q)**.

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/dL),

4. percentage time in range secondary (CGM >= 70 mg/dL and CGM <= 150 mg/dL),

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 intervals: daytime (6 am to midnight), c am), and whole day (12 am to 12 am).

Percentage is with respect to the total number of CGM data that should be available each day. Assume that the t data that should be available is 288. There will be days such that the number of data available is less than 288, b percentage to be with respect to 288.

You have to extract these metrics for each day and then report the mean value of each metric over all days. Hen 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 of the data collection whereas the la: of the data collection. The data starts with manual mode. Manual mode continues until you get a message "AUT(PLGM OFF" in the column "Q" of the InsulinData.csv. From then onwards Auto mode starts. *You may get multiple ACTIVE PLGM OFF" in column "Q" but only use the earliest one to determine when you switch to auto mode. Th back to manual mode,* so the first task is to determine the time stamp when Auto mode starts. **Remember that tl CGM data is not the same as the timestamp of the insulin pump data because these are two different dev asynchronously.**

Once you determine the start of Auto Mode from InsulinData.csv, you have to figure out the timestamp in Auto mode starts. This can be done simply by searching for the time stamp that is nearest to (and later tl

start time stamp obtained from InsulinData.csv.

For each user, CGM data is first parsed and divided into segments, where each segment corresponds to a day w
is considered to start at 12 am and end at 11:59 pm. If there is no CGM data loss, then there should be 288 samp
The segment as a whole is used to compute the metrics for the whole day time period. Each segment is then div
segments: daytime sub-segment and overnight subsegment. For each subsegment, the CGM series is investigat
number of samples that belong to the 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 particular may not have all 288 data points. In
are represented as NaN. You need to devise a strategy to tackle the missing data problem. Popular strategies inc
entire day of data, or interpolation.

Write a Python script that accepts two CSV files: CGMData.csv and InsulinData.csv and runs the analysis proced
metrics discussed in the metrics section in another CSV file using the format described in Results.csv.