

Charles Clevenger
Object Oriented Programming for Mathematics
Professor Xiang Wan
May 3, 2024

T1D Chatbot Gen1

Background:

Type 1 diabetes is an auto-immune disease in which the pancreas fails to produce insulin. Insulin is an essential hormone to process glucose in the bloodstream into nutritionally bio-available fuel for the body. Since type 1 diabetes necessitates that those affected self administer insulin before each meal; type 1 diabetics must properly calculate insulin doses depending upon current blood glucose and planned carbohydrate intake. Modern medicine has advanced the management of T1D with continuous glucose monitors from brands such as dexcom and libre. Trend recognition software by these companies allows users to easily see patterns in glucose levels to make adjustments to their treatment regimen. Using this data in the csv file format I have created a glucose analysis algorithm through python that takes data over a unlimited time range and returns the average glucose, percentage of time in range, and the most common time for spikes and dips, along with the aforementioned graphical depiction of the blood sugar variation with glucose plotted over time. Incorporating this into a chatbot with advice for low and high blood sugar, insulin dosing calculations, a nutritional database, and an emergency hotline I hope to simplify T1D management for myself and others with T1D.

Functionality:

The chatbot has 6 functions which I will discuss below:

1. Low blood sugar
 - Input 1 responds with the message, "If you are experiencing low blood sugar, take 15 grams of sugars and wait 15 minutes." which is appropriate and common advice for moderately low blood sugar.
2. High blood sugar
 - Input 2 responds, "If you are experiencing high blood sugar, a general guideline is to take 1 unit of insulin for every 50 mg above range. Consider drinking water or taking a walk to help lower sugars." which is appropriate and common advice for elevated blood sugar levels.
3. Insulin dosing
 - Input 3 prompts the user to enter their insulin to carb ratio, carbs that will be consumed, and current blood sugar and calculates the required insulin dose for the upcoming meal.
$$\text{insulin_dose} = \text{carbs_in_meal} / \text{insulin_to_carb_ratio}$$

Adjust insulin dose based for blood sugar level

```

if blood_sugar > 100:
    # For every 50 mg above 100 mg/dL, add 1 unit of insulin
    insulin_correction = (blood_sugar - 100) / 50
    insulin_dose += insulin_correction

elif blood_sugar < 100:
    # For every 50 mg below 100 mg/dL, subtract 1 unit of insulin
    insulin_correction = (100 - blood_sugar) / 50
    insulin_dose -= insulin_correction

```

4. Nutrition database
 - Input four prompts the user to enter the name of the food item and returns the nutritional information if it is included in the database.
5. Emergency
 - Input 5 returns the National Diabetes Emergency Hotline at 1-800-XXX-XXXX for immediate assistance.
6. Glucose Analysis
 - Input 6 returns average glucose, percentage of time in range, and the most common time for spikes and dips, along with a graphical depiction of the blood sugar variation with glucose plotted over time.

Module usage:

- Sqlite3: This is used as a database engine this script uses for storing and calling food nutrition.
- Pandas: Used for reading csv files into a data frame, converting column to date time format and numeric format, and dropping missing values, calculating the mean, and providing data frame for plotting.
- Matplotlib.pyplot: For plotting glucose values read from csv.

Functions:

- print_logo(): This function prints the logo of the chatbot.
- percentage_time_in_range(df, lower, upper): Calculates the percentage of time spent in a specified glucose range.
- analyze_glucose_data(): Analyzes glucose data from a CSV file, calculates various metrics, plots the data, and prints the analysis results.
- setup_database(): Creates a SQLite database table and inserts sample food nutritional data.
- respond_to_input(input_number): Responds to user input based on the selected option.
- calculate_insulin_dose(): Calculates the insulin dose based on the user's input of insulin-to-carb ratio, carbs in the meal, and current blood sugar level.

- `get_nutritional_info(food_name)`: Retrieves nutritional information for a specific food from the database.
- `format_nutritional_info(food_name, food_info)`: Formats the retrieved nutritional information for display.
- `save_chat_history(chat_history, file_name)`: Saves the chat history to a text file.
- `main()`: The main function that orchestrates the chatbot interaction. It displays the menu, processes user input, and manages the conversation flow.

Challenges:

After successfully creating the initial graph to interpret and display the csv file I had to make this a function within the chatbot. Next instead of using regular expressions I decided that a '1-6' menu would be the most user friendly option to select from the chatbot functions. To increase user friendliness I also made sure to re-prompt the user to choose from 1-6 or 'quit' or press 0 to reiterate the menu. Finally, upon quitting I wanted the file to be sent to email through Gmail, but Google stopped allowing third party sign-ins in 2022 so this was no longer an option, so I used the global variable `chat_history` to append the inputs and outputs to save to a user named text file upon quitting.

Inspiration:

I drew inspiration from Dexcom's Clarity, an advanced glucose analysis program, and the simple chat-bot created earlier in a university course I took: Object-Oriented Programming. I also utilized: <https://patorjk.com/software/taag/#p=display&f=Graffiti&t=Type%20Something%20> as a command line logo generator for the 'T1D Chatbot Gen1' logo printed at the beginning of the program. <https://docs.python.org/3/library/sqlite3.html> was a helpful resource for the utilizing the sqlite3 module for the food database.

<https://www.w3schools.com/python/pandas/default.asp> was a helpful resource for the pandas module used to read glucose data from the csv file and format for plotting.

<https://matplotlib.org/stable/tutorials/pyplot.html#sphx-glr-tutorials-pyplot-py> Provided helpful information on how to plot glucose values.