Project report

Arizona State University
Course code: CSE 572 - Spring 2021
Course Title: Data mining

Submitted by
Thien Tran
ID: 1222245570
Ira A. Fulton Schools of Engineering

Submitted to
Masudul Quraishi
Lecturer of Engineering
School of Computing, Informatics, and Decision Systems Engineering

Date of Submission:
Mar-05-2021

Table of contents

1. Introduction

Case analysis

Understanding the Medtronic 670G system is crucial to accomplishing the project requirements. According to Dr. Banerjee's project requirements (Programming Assignment 2021), it measures a patient's blood glucose every five minutes via two components: a continuous glucose sensor (CGM) and a Guardian Sensor(GS) (Process P1, appendix A). It stores the patient's data in the CGMdata.csv file (Datastore D1, appendix A). During a meal, the patient enters the estimated carbohydrate intake amount into Bolus Wizard (P2, appendix A). Bolus Wizard input the intake amount along with the other adjusting factors. It stores the data in the Insulindata.csv file (D2, appendix A).

Business requirement

The assignment requires analyzing data and building a machine learning model which uses the sensor glucose to predict the times when a patient takes meals (P3,P4 appendix A). Once the analysis is done, the model is submitted for feedback from the Autograder which is the evaluation software. It uses a test sample to measure the predictive power of the machine (P5, appendix A). A percentage grade is sent back to the machine, the objective function of the assignment is grade percentage maximization. Based on the grade percentage feedback received, the model learns to improve.

Technical requirements

Technology requirements

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

Data requirements

The estimated carbohydrate intake amount is extracted from column Y of CGMdata.csv and CGM_patient.csv (BWZ Carb Input (grams), appendix B).

The sensor glucose amount is extracted from column AE of Insulindata.csv and Insulin_patient2.csv (Sensor Glucose (mg/dL), appendix B).

The dates and times in columns B and C between CGMdata.csv and Insulindata.csv and between CGM_patient.csv and Insulin_patient2.csv must be merged to the closest ones (Date, Time, appendix B).

Predicting machine requirements

The train.py file uses the sensor glucose amount in Insulindata.csv and Insulin_patient2.csv to train a predicting machine(D2 to P3, appendix A). The machine is stored in a picked file. The machine loaded into the test.py file (P4, appendix A) takes a non-label Nx24 matrix (Feature matrix, appendix B) in test.csv (P5, appendix A) to predict whether the meals are taken . It will output the Nx1 vector result (Label, appendix B) in the result.csv (D5, appendix A) file stored in the same path. The format of the output is a Nx1 vector The vector contains only binary numbers.

Submission requirements

The zip file submission must contain only train.py and test.py.

2. Explanation of the solution

Since the assignment is a data analysis, the standard procedures include: data extraction, data preprocessing, data processing, exploratory data analysis and model selection (Appendix C).

The agile methodology is suitable for the assignment design because it is good for projects which have uncertain outcomes, low risks and a scalable property. By repeating the entire data analysis process multiple times, each iteration improves the machine learning model incrementally. The experience gained from the last iteration can be reapplied to the next one and avoid any mistakes from the last ones (Appendix D).

Data extraction, preprocessing and processing

Extraction

The data come from the zip file named Project2Files.zip in the coursera portal. It is manually downloaded and stored into a folder r'ProjectCSE572/assign2'. Using API pandas.read_csv, the local python machine imports the datasets into Pandas dataframes (Pandas.read_csv) .

Dimensionality reduction

The imports only keep date, time and glucose sensor attributes in the CGMdata and date, time and BWZ Carb Input attributes in the Insulindata file. One requirement leads to the merger of the CGM dataframe and the insulin dataframe. The merger uses API pandas.merge_asof to combine every pair of approximate times (time in CGMdata and time in Insulindata) with a minor difference of 5 to 30 seconds (Pandas.merge_asof).

Feature creation

Meal attribute

The meal attribute is the assignment's target attribute. It needs to be created based on the insulin dataset. The guidance in the instruction videos from Dr. Ayan Banerjee speeds up the data processing step. Once the two dataframes are merged, Pandas shift command is performed on the newly merged dataframe (df1) on the BWZ Carb Input (grams) column (reference). The command creates 30 new temporary columns (namely carb -6 to carb 23) with the 5 minute interval each column, and they are from 30 minutes before a reported meal to 1 hours 55 minutes after the reported meal. A new column called meal is created. Each row in the meal is marked as 1 if and only if the new carb 0 attribute value is greater than 0 and all other 29 attribute values are NaN or 0. Otherwise, the meal attribute value is NaN. All of the 30 temporary columns are dropped afterward (Appendix E).

A new column named no_meal is created. The no_meal column is equal to 1 only if the BWZ Carb Input at time 0 is NaN or 0 for the next 2 hours and if no_meal is equal to 0

for the last 2 hours when running through the time series from the beginning to end. Attribute value of meal is set to 0 if no_meal is 1; then no_meal column is dropped (Appendix F).

5 minute interval attributes

According to Dr. Ayan Banerjee instructions, shift function is performed on the insulin column to obtain the new 24 columns (from insulin 0 to insulin 23). Each new column indicates the amount of glucose sensor from time 0 to 115 minutes (23*5).

First order derivative and second order derivative

The first order derivative is computed by successively subtracting column n from column (n-1) of the 24 5-minute interval attributes. Variable n starts from 23 and decrements by 1 until n is equal to 0. By using a similar algorithm, the second order is derived from the first order. The maximum value of each of the 2 new features among them in each row are taken as the feature values.

Exploratory data analysis

Because grade maximization is used as the objective function, rather than exploring the data to find the most optimal solution, the strategy used is the brute force method by way of trying out all possible combinations of available attributes and machine learning algorithms to maximize the return on the grade percentage. The 24 5-minute raw attributes previously mentioned, the first and second order derivatives are combined with decision tree, knn, multilayer perceptron (MLP) and the majority voting machine.

3. Description of the results

Model selection

At the first submission attempt, the first and second order derivatives are selected as the main features. The default decision tree algorithm is used to predict the label attribute (1.10. decision TREES). The submission yields 89% of the grade from the autograder.

At the second attempt, the knn model is used (sklearn.neighbors.kneighborsclassifier). By an iterative process, k (number of neighbors) is optimal at 7 where the change in error is minimal. Nevertheless, the submission yields the same 89% of the grade from the autograder. Discussing about the algorithms, features and methods on the Slack channel spring_a_2021_cse_572_data_mining with the fellow CSE 572 students. The MLP perceptron is added to the model portfolio. At the third attempt, MLP is trained by the 24 raw attributes (neural network models). This submission yields a result of 94%. The last attempt is the joint algorithm namely majority voting of the previous decision tree algorithm, Knn and MLP (Sklearn.ensemble.VotingClassifier). The submission yields the final result of 96%.

Up to the last submission, the project consumes about 24 hours of human labour. Preliminary analysis indicates that roughly to obtain 100% from 96%, the project needs 8 extra hours of human labour. Taking into consideration the future time spent on the next project 3 worth of 20%, the exam 2 worth of 15%, the student in this assignment deems the 8 hour human labour spent on the project 3 would yield more percentage grade than those potentially spent on the current project. Thus, the current project is concluded with a 96% satisfactory result.

4. Description of your contributions to the project

This project is an individual assignment. I do most of the work. That being said, I can't be thankful enough of those who contribute in the slack discussion forum spring_a_2021_cse_572_data_mining. I am especially grateful for these individuals:

- Vladislav Kryshtanovskiy - reminds us that the MLP can yield a superior result.
- Song - shows us how to use the Python pickle package.
- Jianjun Zhang, Song, Ashutosh Joshi and instructor Masudul Quraishi - help me to solve the autograder issue.

5. Lessons learned

Data science methodology

Research shows that systemic thinking results in 40% more efficiency (Abrahams, 2014). I have the chance to incorporate the data science framework learned in the lectures into this project (appendix C). The framework reduces my time in brainstorming the project, framing the problem and finding a suitable approach to the problem.

Working through the project, I improve my expertise in each component of the data science framework: extraction, preprocessing, processing, analyze and select. The framework will help me to tackle any new data science problem in a systemic approach.

Collective communication

Discussing in the slack channel spring_a_2021_cse_572_data_mining helps me to realize what we can potentially achieve as a group or society through communication of ideas and knowledge. As a group, we improve each other's strengths and weaknesses. The channel creates an effective learning environment that everyone contributes to the common goal. We individuals synergize to exponentially create societal values to the class rather additively. With the growth of big data and hyper integration, we advance so much faster than where we did in the last 5000 years of civilization combined.

Coding expertise and overestimation

As someone who graduated from a business school, my tasks are to strategize, design and decide when the business information system is involved. The project opens a new horizon to me. I learn about the information system from the technical perspective. Coding takes longer than what I expected. At the beginning of the project, the total number of hours I allocate for the project is 10. Although the designed solution seems simple, the actual implementation cost 14 hours more than the number budgeted. The cause is my novelty in coding and understanding computer algorithms. For the near future, I have to significantly increase my budgeted hours for the coding tasks. In the long-term, I plan on increasing my total throughput for coding skills and improving my average coding speed.

6. References

Abrahams, M. (2014, December 4). Think Fast, Talk Smart: Communication Techniques. Retrieved March 03, 2021, from https://www.youtube.com/watch?v=HAnw168huqA

Banerjee, A. (2021, Spring). Programming Assignment. Retrieved March 04, 2021, from https://www.coursera.org/learn/cse572/programming/5rhNt/extracting-time-series-properties-of-glucose-levels-in-artificial-pancreas

1.17. neural network models (supervised). (n.d.). Retrieved March 04, 2021, from https://scikit-learn.org/stable/modules/neural_networks_supervised.html#multi-layer-perceptron

Pandas.merge_asof. (n.d.). Retrieved March 04, 2021, from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.merge_asof.html

Pandas.merge_asof. (n.d.). Retrieved March 04, 2021, from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.merge_asof.html

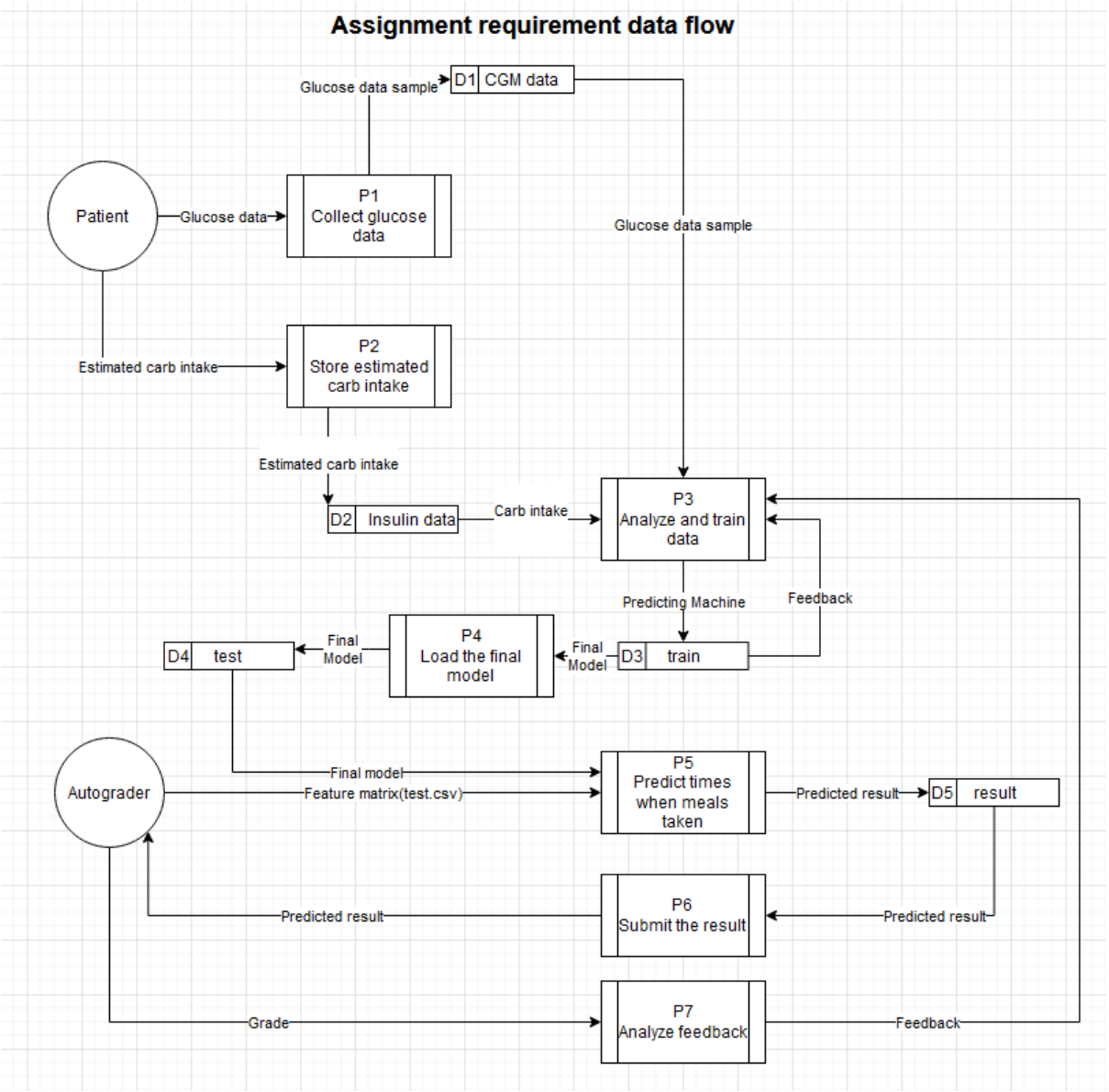
Pandas.read_csv. (n.d.). Retrieved March 04, 2021, from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

Sklearn.ensemble.VotingClassifier. (n.d.). Retrieved March 04, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

Sklearn.neighbors.kneighborsclassifier. (n.d.). Retrieved March 04, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

7. Appendices

Appendix A: Assignment requirement process data flow diagram



Appendix B: Metadata of the assignment

Dataset: CGMdata.csv and CGM_patient.csv

Name	Type	Description
BWZ Carb Input (grams)	Float	The estimated carbohydrate intake amount measured by a unit of gram.
Date	dt(mm/dd/yyyy)	Date of the record
Time	t(hh/mm/ss)	Time of the record

Dataset: Insulindata.csv and Insulin_patient2.csv

Name	Type	Description
Sensor Glucose (mg/dL)	Float	The average amount of glucose in the patient's blood during a 5 minute interval
Date	dt(mm/dd/yyyy)	Date of the record
Time	t(hh/mm/ss)	Time of the record

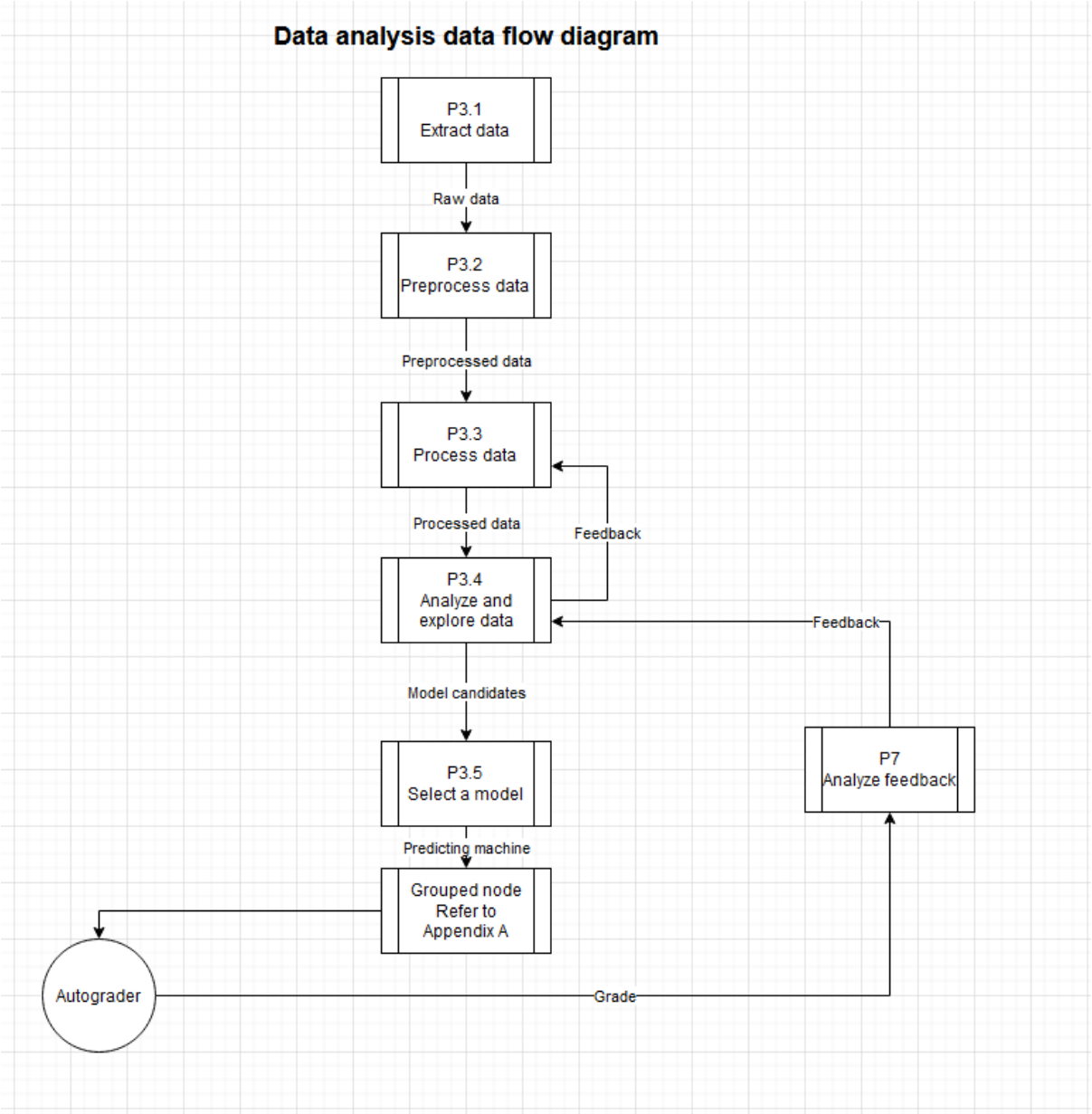
Dataset: test.csv

Name	Type	Description
Feature matrix (1,...,24)	Float	Matrix consists of 24 columns equivalent to 2 hours interval for each record

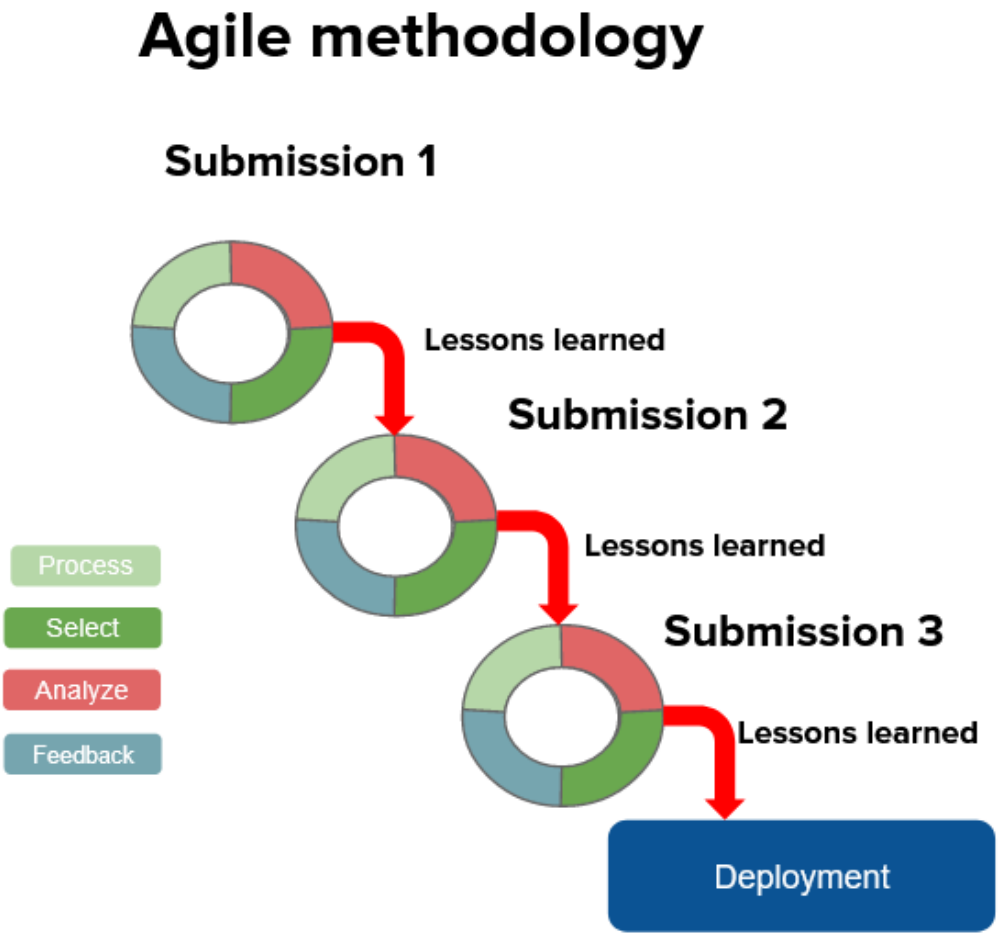
Dataset: result.csv

Name	Type	Description
Label	Binary(0/1)	Represents whether a meal is taken (1) or not(0)

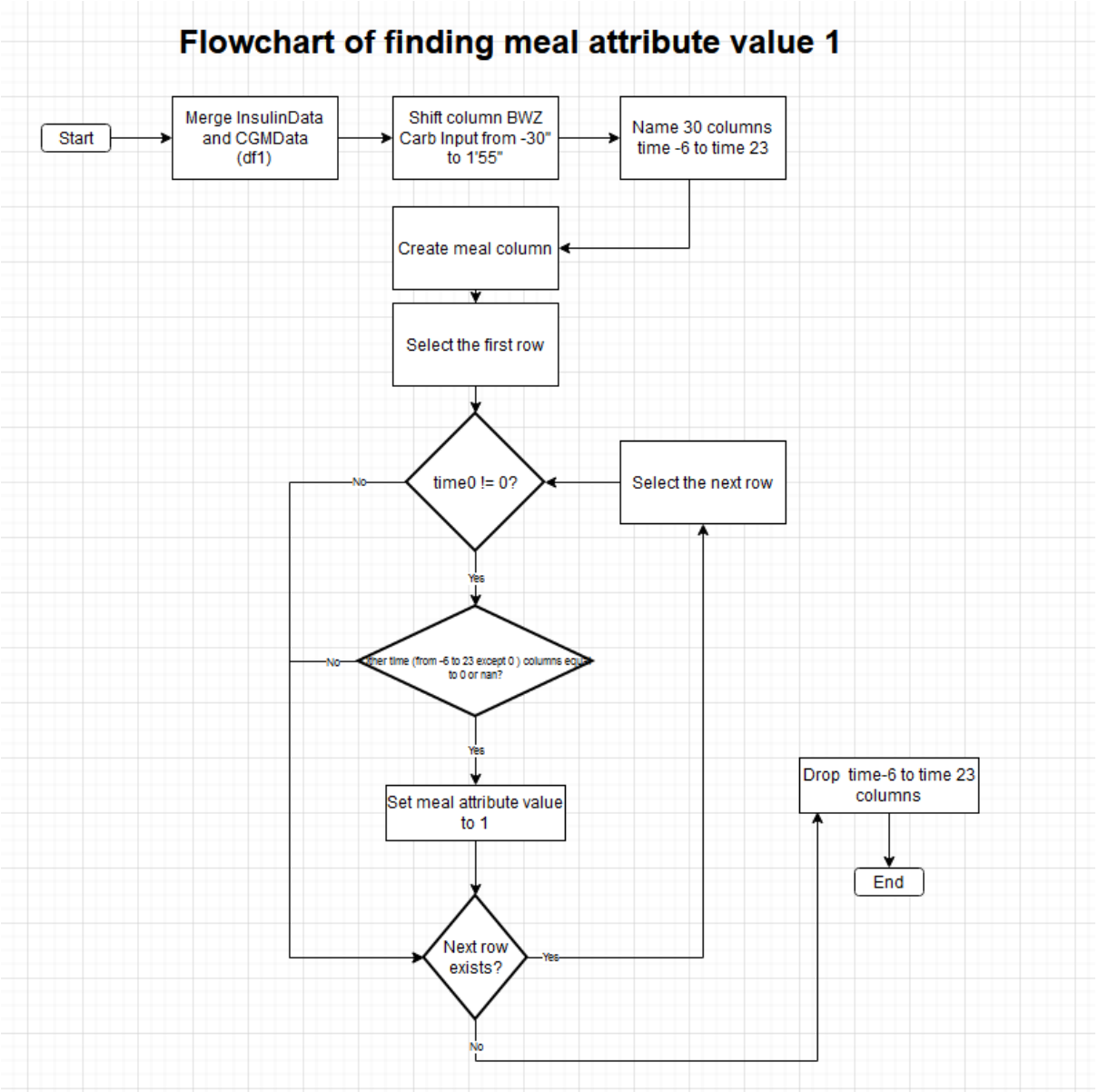
Appendix C: Data analysis data flow diagram



Appendix D: Agile methodology



Appendix E: Flowchart of finding meal attribute value 1



Appendix F: Flowchart of finding meal attribute value 0