CSE 572 - Data Mining
Portfolio Report

Devesh Yadav, 1217208161

1. Introduction

This course required the students to work on 4 projects/assignments (3 mandatory and 1 optional). These assignments required students to write code for programs that performed various tasks, such as - knowledge extraction, anomaly detection, machine learning solutions, association rule mining etc. . All projects combined account for 60% of the course grade. For all assignments we use the data set generated by the Artificial Pancreas medical control system (Medtronic 670G system). This system consists of a continuous glucose monitor (CGM), i.e. the Guardian sensor which is used to collect blood glucose measurements every 5 minutes. For some assignments we use the data collected from multiple patients that use the above mentioned system. The purpose of these assignments were to solve real-world problems related to the artificial pancreas medical control system.

For Assignment 1, we were required to utilize the 'CGMData' and 'InsulinData' to extract the percentage time a patient spends in various levels of hyperglycemia; this data extracted for both day and night. Assignment 2 required development of an end-to-end machine learning solution that could identify if a given set of data points corresponds to meal data or not. Assignment 3 builds upon the feature extracted in the previous assignment; it required us to cluster meal data points based on the amount of carbohydrates in that meal. For Assignment 4, the aim was to determine anomalous events through association rule mining.

2. Description of solution

Before discussing the solution to the assignments in this course - it is important to understand the underlying system that is the source for all our data. As mentioned in the introduction section of this report we are working with Medtronic 670G system which is a type of artificial pancreas media control system. An artificial pancreas is an insulin pump that is connected to CGM (continuous glucose monitor); based on reading from the CGM insulin is delivered to the patient [1].

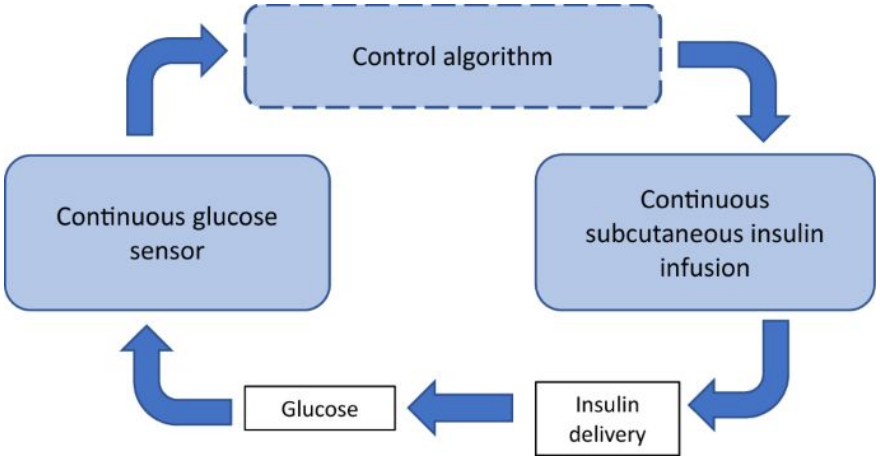

Figure 1 - Overview of an artificial pancreas.

For each patient that utilizes the above-mentioned system, we receive two datasets - from the continuous glucose monitor, and from the insulin pump.

The assignments for this course are designed in such a way that they build upon each other. Therefore, the work done over all assignments can be divided into following parts -

1. Joining CGM and insulin data sets. **(Assignment - 1, 2, 3, 4)**
2. Generating meal and non-meal data points from the combined dataset. **(Assignment - 2, 3, 4)**
3. Generating features from meal and non-meal data points. **(Assignment - 2, 3)**
4. Train supervised machine learning algorithms to predict meal and non-meal data points. **(Assignment - 2)**
5. Train unsupervised machine learning algorithms (such as Kmeans, DBSCAN) to cluster meal data points on the basis of amount of carbohydrate in the meal. **(Assignment - 3)**
6. Detect anomalous events using association rule mining. **(Assignment - 4)**

The process of combining and extracting relevant data from the two data sets is different for assignment 1 as compared to the rest of the assignments. In assignment 1, we search for the first occurrence of a specific keyword in column **'auto mode exit events and unique codes representing reasons'** of our insulin data. The timestamp associated with this keyword is translated to the corresponding timestamp in CGM data. For the purpose of assignment 1 - we consider data points that chronologically follow our translated timestamp to be part of 'auto-mode' and the remaining data points to be part of 'manual-mode'. We extract values from the column **'Sensor Glucose (mg/dL)'** and do basic frequency count for day and night time-period of each day.

For generating meal and non-meal data points, we use the column **'BWZ Carb Input (grams)'** in our insulin data; all non-zero and non-nan values in this column are

considered as possible meal start dates. All meal start dates/timestamps are translated to corresponding time-stamps (i.e. timestamps that immediately follow the meal start date/timestamp in insulin data) in our CGM data. A 2hr 30min time period is carved around the translated timestamp as 'meal-time', all data points within this time period are considered as meal data points. Similar guidelines are provided that are used to find non-meal data points. Because of the difference in the time-period considered for both meal (2hr 30min) and non-meal (2hr) time there is a imbalance in the number of data points. To overcome this imbalance we extract meaningful features from these data points. Some of the extracted features are mentioned below -

1. CGM_max - CGM_min : Difference between highest and lowest values for column 'Sensor Glucose (mg/dL)' for a meal or non-meal.
2. time_CGM_max - time_CGM_min : Difference between the time when the highest and lowest values occur for column 'Sensor Glucose (mg/dL)'. This is a characteristic of the human body which is known as 'Meal absorption time'.
3. CGM_Velcoity
4. 2 dominant frequencies extracted using fast fourier transform
5. Windowed mean of the data points for meal or non-meal data.

For supervised learning and unsupervised learning, popular python machine learning library 'scikit-learn' was used to train and test various algorithms. To evaluate the performance of algorithms like DBSCAN and KMeans some metrics like purity, entropy and SSE were implemented from scratch. For association rule mining, 'mlxtend' was used to extract most frequent sets, rules with highest confidence, and anomalous rules.

3. Results

For Assignment 1, the observed results are displayed in the form of a char below.

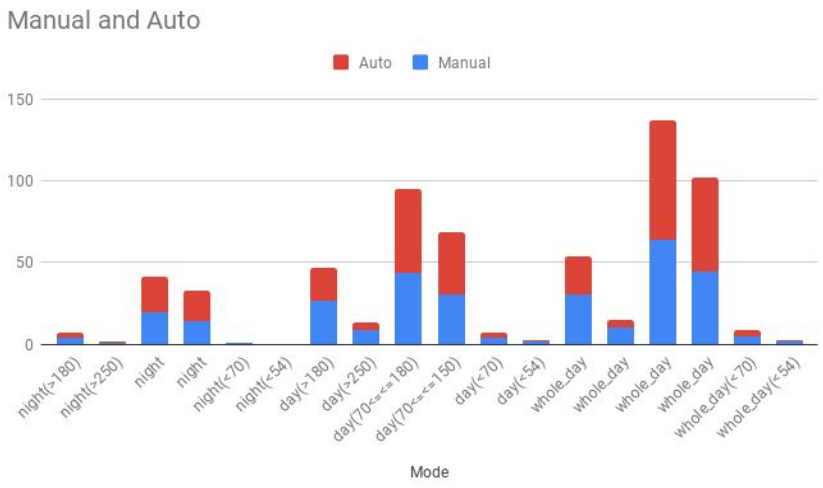

Figure 2 - Observed results from assignment 1.

For Assignment 2, various machine learning algorithms like SVM, decision tree, neural network, etc were trained and tested. Random forest produced the best results with an accuracy of 91%. Performance of the model was validated using KFold cross validation. In assignment 3, only KMeans and DBSCAN algorithms were used to perform clustering. Their performance is shown in figure 3.

{'fit_time': array([0.4406414 , 0.29283571, 0.4665916]), 'score_time': array([0.02203202, 0.02418303, 0.0330584]), 'test_f1': array([0.79835391, 0.88235294, 0.832]), 'test_recall': array([0.73484848, 0.82317073, 0.76470588]), 'test_accuracy': array([0.90429688, 0.9295499 , 0.91780822]), 'test_precision': array([0.87387387, 0.95070423, 0.9122807])}

Figure 3 - Metrics for Random forest algorithm.

SSE_Kmeans	SSE_DBSCAN	Entropy_Kmeans	Entropy_DBSCAN	Purity_Kmeans	Purity_DBSCAN
539.57	2052.97	2.065086526	0.547417438	0.34	0.91

Figure 3 - Metrics for KMeans and DBSCAN.

4. Lessons Learned

The lessons I learned during the course of this project are as follows -

a. **Modular Code -** As mentioned in previous segments, the assignments built on top of each other. There was a lot of code that was reused with little to no modification. Making the code modular enabled me to tackle the assignments very fast.

b. **Documentation -** A well documented code saved me a lot of time while doing the assignments. Some tasks only required little modification on previously written code. Documentation helps revisiting the code much easier.

c. **Data Cleaning and Feature Extraction -** As expected from the course 'Data Mining' - data cleaning and feature extraction proved to be the most important and significant part of the assignments. I revisited this step multiple times during each assignment. A simple change in the meal data point extraction step helped improve my results and achieve perfect score in the assignment.

5. References

[1] Artificial Pancreas - https://www.fda.gov/medical-devices/artificial-pancreas-device-system/what-pancreas-what-artificial-pancreas-device-system

[2] Scikit-learn - https://scikit-learn.org/stable/