Learning Predictive Models of Meal Behaviour

Nishtha Kalra

Submitted for the Degree of Master of Science in

Machine Learning

Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20 0EX, UK

September 3, 2019

Declaration

This report has been prepared on the basis of my own work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count: 8508

Student Name: Nishtha Kalra

Date of Submission: September 3, 2019

Signature:

Acknowledgement

I am grateful to my supervisor Dr Nicola Paoletti for his expertise and advise throughout the making of my project. He guided me and helped me out whenever I was in any doubt. I would also like to thank my university, Royal Holloway University of London, for giving me an opportunity to study in this esteemed university. I would also express my gratitude to my teachers for teaching me each concept of machine learning with utmost sincerity and expertise.

I would especially like to thank my parents and my brother for their continued support. Also, I would like to thank Akshay for proofreading my report and my friends for giving me the strength to complete this project. In the end, I would thank God for always being there and showing me the right path in life.

Abstract

Diabetes is a condition in which the blood glucose level of a patient increases more than the normal level. This happens because of two reasons - when enough insulin is not produced by the body to move the glucose or when insulin produced, is destroyed by the body's immune system. To cope with this condition, patients inject insulin in their body either by insulin pump or by insulin pen. Care needs to be taken that the right amount of insulin is injected in the body. An incorrect amount of insulin injection can result in hyperglycemia and hypoglycemia. Meal announcements need to be made by the patients to determine the amount of insulin needed to be injected. When a patient uses artificial Pancreas, the insulin level is automatically and intelligently monitored and adjusted.

This project aims at applying machine learning methods to predict meal behaviour of all participants in order to predict their future blood glucose level and then use this information to make better insulin decisions. A time series is created for time against carbohydrates, which helps in solving this issue. The data is split into many clusters by different methods so that similar data are grouped together, and thus, prediction can be made easily on such groups.

Contents

1 Introduction

According to the National Diabetes Statistics Report, 2017 [9], more than 100 million US adults are either diabetic or pre-diabetic. It is a very critical condition which if not treated properly, can cause damage to a person's life. When a person's body is not able to either produce insulin or is not able to use the insulin produced, then it results in increasing the blood glucose level and thus in diabetes. In such cases, patients inject insulin externally inside the body. Hyperglycemia and hypoglycemia are one of the significant problems related to diabetes, which can occur if insulin intake is not monitored correctly.

Numerous advancements are being made today to make a patient's life more comfortable to deal with diabetes. Machine learning allows us to develop algorithms and techniques that allow computers to learn and predict results after analyzing previous information or data. Machine learning has reached enormous heights now that people have started relying on it for crucial future predictions.

Artificial pancreas has gained popularity from a couple of years now. They are being increasingly used by patients of Type 1 Diabetes. The glucose monitor detects the blood glucose level in the body. The insulin pump delivers insulin to the body so that carbohydrates are absorbed by the cells. A control algorithm is responsible for communication between the glucose monitor and insulin pumps, which collects data from the glucose monitor and informs the insulin pump when and how much insulin needs to be released in the system.

Our aim of this project is to determine a method which can predict carbohydrates levels, at a particular time, given the past instances of time and carbohydrate level. However, the issue is the eating pattern of people keeps on changing subject to activities like stress, exercise, diets, health and many more.

In hybrid closed-loop device, patients need to announce the time and amount of carbohydrates intake manually at each time. In case a patient makes a mistake in announcing the information, an incorrect insulin dosage will be injected in his/her body, and blood glucose levels can be disrupted harmfully [16]. To remove such risks, meal predictions are needed, which will automate this procedure and will eliminate the task of meal announcements by the patients.

A flowchart is depicted in Figure 1, which shows the flow of this project.

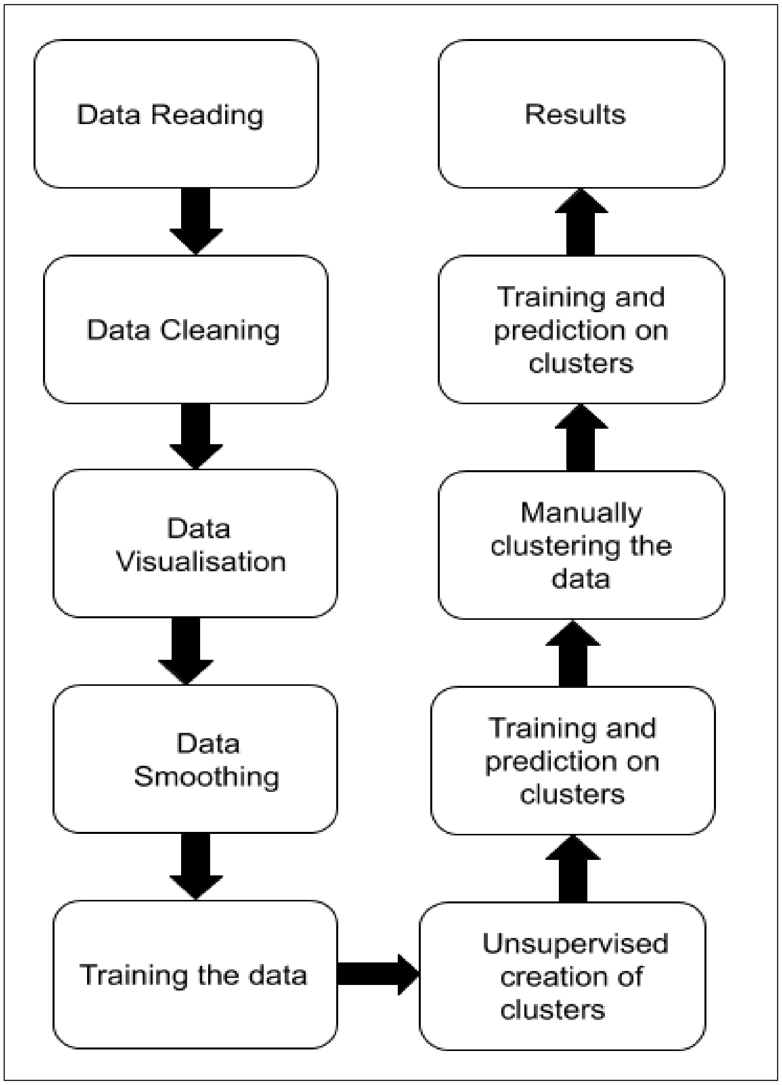
Data Reading
Data Cleaning
Data Visualisation
Data Smoothing
Training the data
Unsupervised creation of clusters
Training and prediction on clusters
Manually clustering the data
Training and prediction on clusters
Results

Figure 1: Flowchart depicting the flow of the project

2 Background Research

2.1 Diabetes

Living organisms require energy to power up their systems. Energy is needed by our bodies to do any simple task ranging from digestion and absorption, exercise, work, play, eat. Even activity as simple as sleeping needs energy. Where does this energy come from? The food we eat is responsible for providing us with this energy. This food provides us with a form of sugar known as glucose. Glucose is the most crucial element needed in the primary health system. The cells in our body are powered by glucose, which gives us the energy to achieve basic tasks of our everyday life.

Glucose mainly comes from food rich in carbohydrates like bread, potatoes etc. As we are eating food, it travels down our oesophagus into our stomach. While it is getting digested, our acid and enzymes present in our stomach break down the carbohydrates from the food to make glucose. Then this glucose goes into our intestines where it gets absorbed. From there, it passes in our blood to reach our cells. As this glucose is travelling from your bloodstream to your cells, this glucose is known as blood sugar.

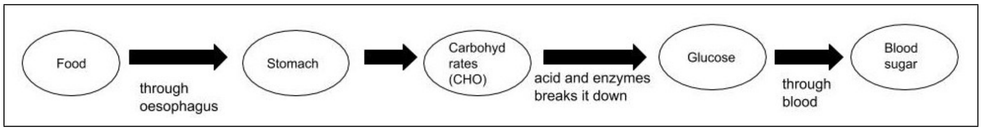

Figure 2: How Glucose is generated in our body

Now, glucose cannot directly reach our cells through blood. It needs something known as insulin. Insulin is a hormone produced by the pancreas. Insulin is like a key to the door of our cells that lets glucose enter. In other words, insulin unlocks muscle, fat and liver cells so glucose can get inside them. Our body keeps the level of glucose in our body constant. The blood sugar level in our body is monitored every few seconds by beta cells present in our pancreas. When we eat carbohydrate-rich food, the blood sugar rises, thats when beta cells, present in the pancreas, releases insulin in our bloodstream.

The glucose that is not used by the body is stored in the liver in the form of glycogen. The amount of glycogen stored in the liver is enough to give energy to our body for one day. Our blood glucose level drops if we dont eat for some hours. There is no insulin produced by the pancreas. That is when a different hormone known as glucagon is produced by the pancreas that lets the liver to break down glycogen into glucose.

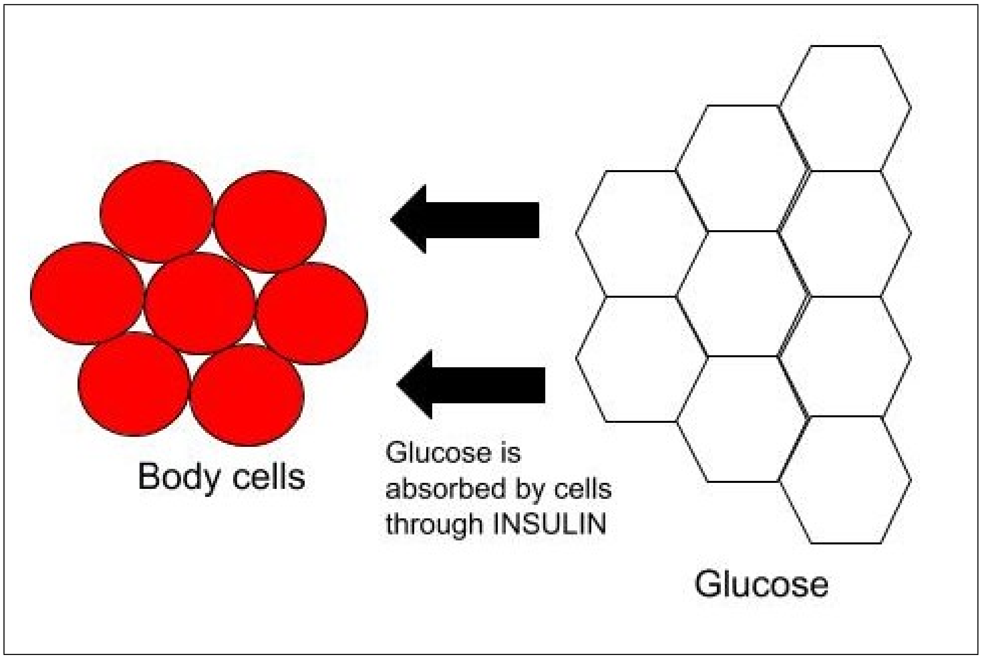

Figure 3: Purpose of Insulin in our body

Diabetes is a condition when our body is unable to break down glucose into energy [21]. The blood sugar level in a persons body is too high. This is because either the insulin produced by the pancreas is not able to work correctly or the insulin produced is not enough. This results in sugar getting accumulated in our blood. Symptoms include frequent urination, loss in weight, feeling tired, all-time thirsty, several infections, wounds dont get healed quickly etc. ultimately damaging our heart, eyes, kidney and feet. When we dont have diabetes, the pancreas makes the correct decision as to when and how insulin needs to be released that can help absorb glucose. But this does not work if we have diabetes.

2.1.1 Types of Diabetes

There are two types of diabetes, Type 1 and Type 2.

Type 1 Diabetes: 1 out of 10 people who have diabetes has Type 1 diabetes. In Type 1 diabetes, insulin is not prepared by the pancreas. Our body attacks the cells responsible for producing insulin; therefore, our body cannot produce it. When there is not any insulin to absorb the glucose present in our blood, more and more glucose gets build up in our blood. Our body tries to get rid of the glucose via kidneys through urination, which results in frequent urine and an extreme feeling of thirst. Our body ultimately feels exhausted because it cant get the energy from glucose. Our body compensates for this loss of glucose by breaking down fats stored in our body and using them for energy. This results in loss of weight.

In healthy people, blood sugar values range between 70-200 mg/dL [16]. But in Type 1, the blood sugar level is very high. According to Diabetes UK, 4.7 million people in the UK have diabetes. 8% have Type 1 Diabetes, 90% are having Type 2, and the remaining 2% have a rarer type of diabetes. We are going to delve down more about the treatment of Type 1 diabetes later.

Type 2 Diabetes: In Type 2 Diabetes, the insulin produced by your pancreas either cannot work properly, or the pancreas cannot make enough insulin. Common synonyms include feeling tried, extreme thirst, frequent urination.

2.2 Treatment of Type 1 Diabetes

Type 1 Diabetes can be dealt with when the patient is provided with insulin, but care needs to be taken to avoid hyper and hypoglycemic episodes. Hyperglycemia occurs when blood sugar levels are too high (140mg/dL). People develop hyperglycemia if their diabetes is not treated correctly. Hypoglycemia sets in when blood sugar levels are too low (60mg/dL). This is usually a side effect of treatment with blood-sugar-lowering medication [11].

2.2.1 Open Loop

In the open-loop method, the patient injects insulin to him/herself at different times of a day. They usually inject insulin in the morning to provide the basal insulin requirement throughout the day. Basal insulin, also known as background insulin, keeps the blood glucose level consistent during fasting. In fasting, the body evenly releases glucose into our blood. Basal insulin is used to keep blood glucose level under control such that the cells can absorb glucose for energy. It is usually taken once or twice in a day. Once injected, it can provide a steady release of insulin all day.

On the other hand, another type of insulin, known as bolus insulin, is specifically taken at mealtimes to keep blood glucose level under control after a meal. Bolus insulin acts quickly on our body. Bolus insulin is usually taken before meals. In some cases, people take bolus insulin during or just after a meal to prevent hypoglycemia. The amount of insulin to inject will depend on both a measurement of glucose and on an estimate of the amount of food that is about to be eaten.

2.2.2 Closed Loop - Artificial Pancreas

Closed-loop insulin delivery is an emerging technology helpful for people who have Type 1 diabetes. It is a device consisting of a continuous glucose monitor, a control algorithm and an insulin pump [16]. The continuous glucose monitor provides glucose measurements after a regular period to the control algorithm. This algorithm is responsible for maintaining healthy blood glucose levels to avoid hyper and hypoglycemia. This control algorithm is running inside the insulin pump. These components together act as a device which regulates insulin intake inside a patients body, thus the name artificial pancreas. Wireless communication facilities automate data transfer between components. But the critical element of the artificial pancreas is the control algorithm [7].

A wearable artificial pancreas closes the loop between a glucose sensor and an insulin infusion pump. This significantly improves the quality of life of diabetic individuals. The involvement of the patient in maintaining glucose control is minimal [8]. Such a system would be able to determine the insulin requirement in real-time, regardless of the situation, and deliver the proper insulin dosage. It would be able to change the infusion as the patients activity changes and, ideally, would exist internally, eliminating the requirement of wearing external equipment. Such a system would also aim to significantly reduce the number of injections required or to eliminate them.

In the case of the artificial pancreas, meals are predicted by using machine learning models instead of being explicitly announced by the patients. In meal announcements by the patient, there is a risk that patient can make a mistake in giving details of the meal, Then an incorrect insulin dosage will be injected in his/her body, and blood glucose levels can be disrupted harmfully. Whereas in the case of meal prediction, this risk is eliminated.

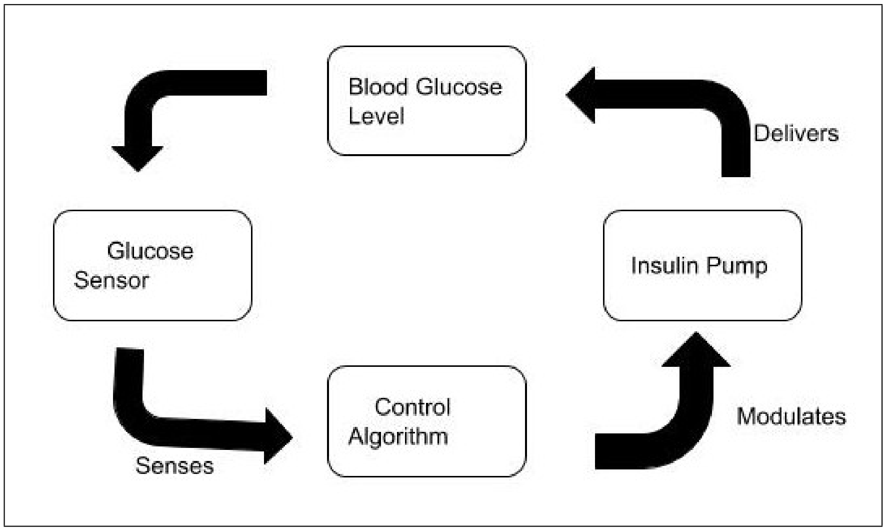

Figure 4: Artificial Pancreas

3 Machine Learning Concepts Used

3.1 Supervised Learning

Supervised learning is a type of machine learning where our task is to create a learning method which will maps an output to an input, given sample inputs and outputs. The sample inputs and outputs are known as training data, which is composed of samples or inputs (X), which have their labels or outputs (y). It is a target function (f) that maps input variables (X) to an output variable (Y).

$$y = f(X) \tag{1}$$

Lets take an example of the iris dataset. The iris dataset is a classical dataset in machine learning and statistics, collected by Ronald A. Fisher [13]. A hobby botanist would like to tell the species of iris flowers that she found. She has a training set of labelled flowers. The features are the length and width of the petals, and the length and width of the sepal, all measured in centimetres. There are three possible labels (species): Setosa, Versicolor, or Virginica. In Iris dataset example,
X = length of petal, width of petal, length of sepal, width of sepal
y = species name(either of Setosa, Versicolor or Virginica)
learning method = a function that maps X to y

The outputs in supervised machine learning problems are of two types. They can either be a discrete class label or a continuous quantity. In classification, the task is to get a mapping function from input variables (X) to discrete output variables (y). The output variables are called labels or categories, and the mapping function predicts the class for a given observation.

In the case of regression, the task is to get a mapping function from input variables (X) to a continuous output variable (y). The output variable is a real-value, such as an integer or floating-point value. A way to solve machine learning problems when there is a regression is linear regression.

3.2 Underfitting and Overfitting

Generally, when a machine learning algorithm gives poor performance, then that is either because of overfitting or underfitting. Overfitting refers to a model that models the training data too well [3].Bias in a learning method tells the amount of assumptions made during training the method. Variance is the estimate of the change in the target function if different training data is used.

Overfitting happens when:

- A machine learning algorithm captures the noise of the data while training.
- The model fits the training data too well. This can be seen when the training score is too high, but the test score is low.
- The model has low bias but high variance.

Overfitting can be solved by fitting multiple models on the dataset. It can also be removed by using validation or cross-validation.

An underfitted model can neither model the training data nor generalize to new data. Underfitting happens when:

- The machine learning method used cannot capture the underlying trend of the data.
- The model used does not fit the data well.
- The model has low variance but high bias.

Underfitting can be solved by trying different machine learning methods. It can also be solved by increasing the size of the training data.

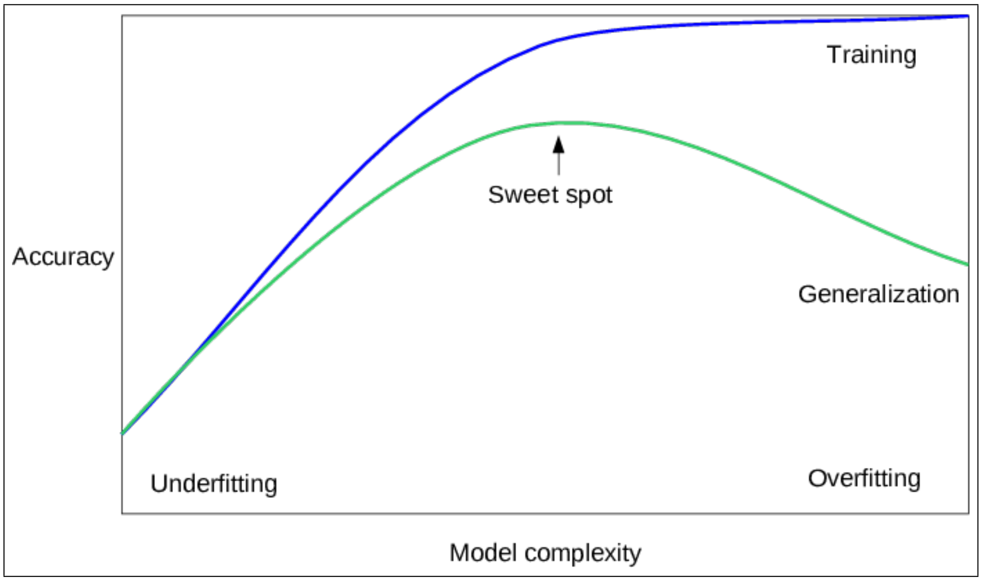

Figure 5: Balance of overfitting and underfitting [24]

Figure 5 explains a balance of overfitting and underfitting in a model [24]. The blue line shows training accuracy, which increases with the complexity of the model, thus leading to overfitting. If the model is not complex, then accuracy is less, which results in underfitting. The green line shows generalization which when decreases in case of underfitting and overfitting. Our aim for a model should be to find a good fit which is indicated as a sweet spot in the figure 5.

3.3 Linear Regression

Linear Regression is an approach to model a relationship between a dependent and an independent variable. When there are more than one independent variables, the process is known as multiple linear regression. It is represented as a linear equation that combines a specific set of input values (x) and provides a predicted solution($\hat{y}$) [1]. The input values and the output value are numeric. The linear equation is made up of coefficients and intercepts. The coefficient in a linear equation is a one case factor assigned to each input value. Intercept is an additional coefficient added to the line to give an extra degree of freedom. The intercept is also known as the bias coefficient.

The equation of the model of a simple linear regression can be written as:

$$\hat{y} = w \cdot x + b \tag{2}$$

where $b =$ intercept and $w =$ coefficient. Dependent variable in this equation is $\hat{y}$ and independent variable is x.

When training a regression model, the coefficients are learned and fitted to training data. The aim is to find the best fit line and minimize the cost function. We try to minimize the error between actual and predicted values. We can measure the error using the cost function [6]. Coefficients are also known as slope. In equation 1, w is the effect on ŷ when x is increased by 1 unit.

When we have more than one input, the line is a plane or a hyperplane. Suppose if we have p features or inputs then equation 1 can be written as:

$$\hat{y} = w_0 \cdot x_0 + w_1 \cdot x_1 + \ldots + w_{(p-1)} \cdot x_{(p-1)} + b \tag{3}$$

where where x_j is the $(j + 1)$st feature and w_j are its coefficient. The dependent variable here is $\hat{y}$ and independent variables are x_0 to $x_{(p-1)}$.

When the parameters are estimated, they are then used for prediction. Please note that in a linear model, we aim for linearity in the parameters; there does not have to be linearity in the attributes.

3.3.1 Least Squares

The parameters need to be estimated to make a prediction in any machine learning algorithm. In the case of linear regression, the parameters w(slope) and b(intercept) are estimated using the approach of Least Squares. We choose w and b such that the Residual Sum of Squares or RSS is minimized [25].

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y_i})^2 \tag{4}$$

where y_i is actual label and $\hat{y_i}$ is predicted label for (i)th data of size n. The Total Sum of Squares or TSS can be calculated using the following formula:

$$TSS = \sum_{i=1}^{n} (y_i - \overline{y})^2 \tag{5}$$

where y_i is actual label for (i)th data of size n and $\overline{y}$ is the average label:

$$\overline{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{6}$$

where y_i is actual label for (i)th data of size$n$.
R^2 is the measure of variability in the label. It is given by the following formula:

$$R^2 = \frac{TSS - RSS}{TSS} \tag{7}$$

where TSS and RSS is calculated using the formula given in equation (4) and (5).

3.4 Ridge Regression

Ridge regression is a technique of regularization when there is overfitting in a model where linear regression is used. In ridge regression the coefficients are chosen such that:

1. Their magnitude is as small as possible so that their effect on prediction is minimum.
2. However, they should be good enough to predict results with reasonable accuracy.

The parameters are coefficients (w) and intercept (b). RSS calculated in equation (4) can also be written as:

$$RSS = \sum_{i=1}^{n} (y_i - w_i \cdot x_i - b)^2 \tag{8}$$

In Ridge regression, w and b are chosen such that the following equation can be minimized:

$$RSS + \alpha \|w\|^2 = RSS + \alpha \sum_{j=0}^{p-1} w_j^2 \tag{9}$$

where RSS is as given in equation (8); p is number of features; w_j is the coefficient corresponding to the feature j and α is the regularization parameter with condition $\alpha \geq 0$.

When $\alpha = 1$ then Ridge Regression acts as Linear Regression. As α becomes bigger, the coefficients gets smaller. This is also known as L_2 regularization.

3.5 Lasso Regression

In Ridge Regression, coefficients become 0 only when $\alpha = \infty$. Otherwise, the coefficients are not 0. Therefore, all the features are used. If the number of features is too large, then using all features can be a problem and might result in overfitting again. Here is where Lasso Regression comes into the picture. In Lasso Regression, many coefficients are set to 0. In a way, we can say that Lasso performs model selection.

In Lasso regression, w and b are chosen such that the following equation can be minimized:

$$RSS + \alpha \|w\|_1 = RSS + \alpha \sum_{j=0}^{p-1} |w_j| \tag{10}$$

where RSS is as given in equation (8); p is number of features; w_j is the coefficient corresponding to the feature j and α is the regularization parameter. This is also known as L_1 regularization.

3.6 Decision Trees

A decision tree is used to create a model that predicts an output by learning decision rules inferred from data. It acts like a tree graph where questions are asked on attributes at its nodes, answers to the question or on edges and leaves have the actual output [12].

It is constructed when partitioning is done recursively. The parent node is split into children left and right nodes, which themselves become a parent node and then split into another left and right node recursively. Figure 6 shows a simple decision tree which gives the smallest of three numbers.

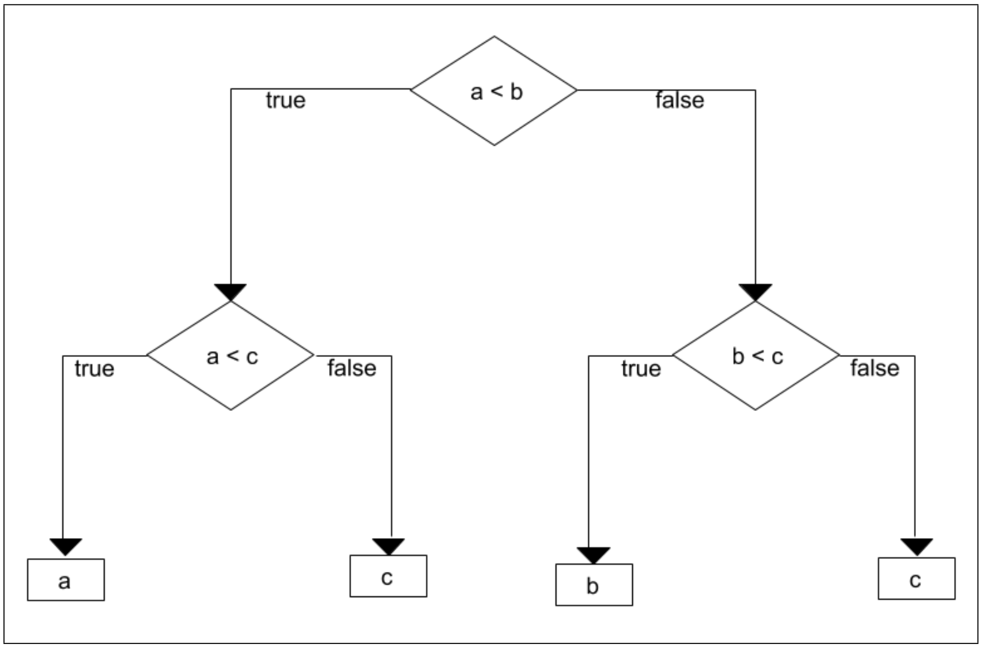

Figure 6: A simple decision tree to select minimum number

Decision trees are created into two steps- induction and pruning [20]. In the induction step, we determine the best feature on which we should prune. We ask a question on this feature. We then split the data into subsets that contain the possible answers to the question raised on the feature. At this stage, will iteratively try out different split points and then at the end select the one that gives us the lowest cost of the cost function. The cost function is generally mean squared error or MSE in the case of recursion. Then a node is created at the splitting point. Recursively, new tree nodes are generated by using the subset of data created.

The next stage is pruning, which helps in cutting the unnecessary splits in our tree. A simple way of pruning is to go through each node in the tree and evaluate the effect of removing it on the cost function.

3.7 Random Forest

Random Forest is a machine learning approach that uses the ensemble method for prediction. In ensemble method of prediction, multiple machine learning algorithms are taken into consideration, and their predictions are combined to make a more accurate final prediction.

Random forest is an ensemble machine learning method which comprises of multiple decision trees. The trees in the random forest run in parallel, and there is no interaction between these trees. Multiple decision trees are constructed while training a random forest, and then the output is a mean prediction of all the individual predictions of the decision trees. It is used to prevent overfitting so that the model does not rely on any single feature. Figure 7 shows an implementation of a random tree which combines prediction acquired by 600 decision trees and averages them to give a final prediction.

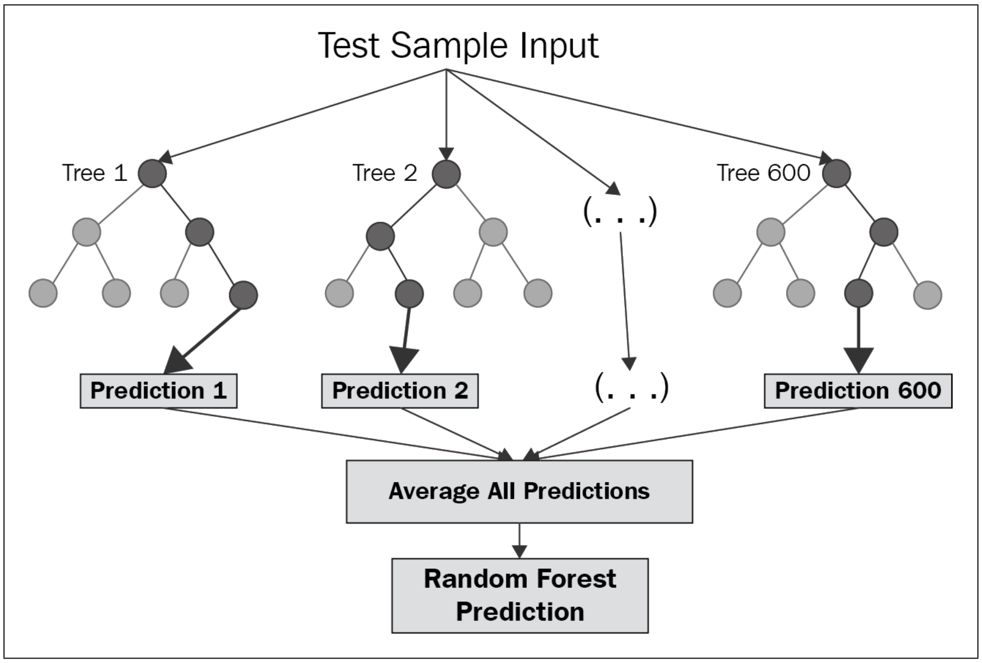

Figure 7: An example of Random Forest [14]

3.8 Unsupervised learning

Unsupervised learning is a type of machine learning algorithm which has inputs or samples, but unlike a supervised learning method, the inputs do not have the associated outputs. Therefore, unsupervised learning is used to find the underlying structure of distribution in the data to learn more about the data. They are unsupervised because there is no teacher to supervise the algorithms and tell them what is correct and what is wrong [4]. Algorithms discover the structure of data on its own without any supervision.

Let us take the example of iris dataset explained above in section (3.1). In the case of unsupervised learning, we have
$X =$ length of petal, width of petal, length of sepal, width of sepal. But we do not have y. One way to solve this problem can be to group samples with similar features together. So the iris data with have similar lengths and widths of petals and sepals are grouped together, and we assume that each of the group or cluster is its own species.

The uses of unsupervised learning can be broadly categorised into:

- finding the underlying structure of a dataset
- grouping the data so that it can be useful
- representing data in a compressed format

The above tasks can be broken down into two methods: clustering and reducing dimensionality.

3.9 Clustering

Grouping of samples in the data in such a way that the samples that are similar to each other are grouped in a cluster is known as clustering. In unsupervised learning, clustering is used to create clusters of similar samples in a dataset. There are various ways in machine learning to create such clusters. These algorithms differ in the way they make clusters. Some algorithms consider the distance between samples as a criterion for clustering; some consider the dense area. We select the appropriate clustering algorithms and corresponding hyperparameters depending on the dataset and result desired.

Clustering is used in Biology to group genes into gene families, in social network analysis to recognise communities and many more such examples. There are two types of clustering methods [26]:

- Parametric Clustering - density estimation with a mixture of gaussian.
- Non-Parametric Clustering - natural grouping or clustering in the dataset.

In non-parametric clustering, first, a measure of similarity or dissimilarity is defined. Then an objective function is defined which measures how well the data is represented in the clusters, and then it is optimised.

3.9.1 K-means clustering

K-means clustering is a type of non-parametric clustering. In K means clustering, the data is divided into K clusters or groups. When K is large, then the number of clusters is more, and then granularity in each cluster is more. When K is small, the number of clusters is less; thus, the granularity is less. In this method of clustering, Input is the dataset and Output is a set of K labels, assigning each data to one of the label.

The method used for clustering is by calculating the centroid of each group. It can be imagined as a party where there are lots of people, and there are some people who become the centre of attraction because they are magnetic. These people act as centroids, and other people gather around them depending on their liking.

The algorithm of K-means algorithm is as follows:

1. Randomly create K centroids for the K clusters.

2. Measure the distance of each data point with each of the K centroid. Usually, Euclidean distance is used to measure the distance. The closeness measure is a hyperparameter here.

3. Assign each data point to the cluster, which has minimum distance with its centroid. For example, if a data point has minimum distance with centroid number 2 then assign this data point a label of 2. This is given by the following equation [19]:
   If c_i is the collection of centroids in set C, then each data point x is assigned to a cluster based on

   $$\min_{c_i \in C} dist(c_i, x)^2 \tag{11}$$

4. Update the centroid for each of the cluster.
   Let the centroid be S_i for ith cluster after calculating it from equation (13). Then the updated centroid of cluster i is:

   $$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i \tag{12}$$

5. Repeat steps 2 and 4 until the algorithm converges.

Silhouette Choosing the value of K or the number of cluster in K-means clustering is very crucial. An incorrect value of K might not group the data points in the dataset properly. There are some ways by which we can validate the clusters formed during K-means clustering. One such method is the Silhouette method.

The silhouette value is a measure of how similar an object is to its own cluster compared to other clusters [18]. The value is between -1 to $+1$. If the value is high, then that means that the data point is well-matched with its cluster and poorly matched with other clusters. A negative or low value of silhouette indicates that the number of clusters is either too less or too more. Choosing the method of distance calculation while calculating silhouette value is a hyperparameter. It can be calculated with any method like Euclidean distance of Mahattan distance.

3.10 Moving Average Smoothing

Moving average smoothing is a technique to remove variations in a time series. It is used for data preparation, feature engineering and prediction. Smoothing is used to remove noise in causal processes.

When a moving average is calculated for a time series, a new series is created which is composed of averages of raw values of original time series. A window size is initialised, which gives the number of observations that should be used to calculate the average. This window slides in the time series to calculate the average values in the new series. There are two ways of calculating moving average namely centered moving average and trailing moving average [2].

3.10.1 Centered Moving Average

In centered moving average, the value is calculated at time t by taking into consideration the past, present and future values. For example, if the window size is three, then the value at time t is calculated as [2]:

$$ma(t) = mean(obs(t-1), obs(t), obs(t+1)) \tag{13}$$

where $ma(t)$ is the centered moving average at time t; $obs(t)$ is observation at time t; $obs(t-1)$ is observation at time $t-1$ and $obs(t+1)$ is observation at time $t+1$. Centered moving average method is used to remove the trend and seasonality of a time series.

3.10.2 Trailing Moving Average

In trailing moving average, the value is calculated at time t by taking into consideration the past and present values only. For example, if the window size is three, then the value at time t is calculated as [2]:

$$ma(t) = mean(obs(t-2), obs(t-1), obs(t)) \tag{14}$$

where $ma(t)$ is the trailing moving average at time t; $obs(t)$ is observation at time t; $obs(t-1)$ is observation at time $t-1$ and $obs(t-2)$ is observation at time $t-2$. Trailing moving average method is used for time series forecasting.

4 Implementation

4.1 Aim

This project aims at predicting meal behaviour in order to predict future blood glucose levels and thus make better insulin decisions. One way to determine meal behaviour is if we can predict the blood glucose level in the body of a person at a given time in future. Given the time and carbohydrates of the previous meal, we can predict carbohydrates at present and thus we can determine the amount of insulin the person needs to inject to fight Type 1 Diabetes. In this way, we will not require meal announcements from the patient, thus removing the risk of hyperglycemia and hypoglycemia.

For implementing this, I have used numpy and pandas, libraries of Python version 3.7.0. To implement different machine learning models, I have used scikit-learn. Scikit-learn is a free software machine learning library for Python.

4.2 Data Cleaning

To achieve our aim, it is necessary to extract time and carbohydrates of all participants from NHANES data and create a time series of it. Data cleaning is an essential step in any data prediction techniques. If data is not cleaned adequately, then we can end up in getting incorrect results.

I loaded and cleaned data in this project in the following way:

1. Downloaded NHANES 2015-2016 dietary interview data from [10] and saved it on my system.

2. Read data in jupyter notebook using padas read_sas function into a dataframe.

3. Drop the rows containing any null values.

4. Create a dataframe with only time and carbohydrates feature copied from the original dataframe.

5. Sort the dataframe by time feature.

6. The time given is in seconds. I converted it into hours to get a better understanding of the series.

7. There were some values which were even less than $5e^{-}20$. Dropped the rows containing such values.

8. Change the index of dataframe from automatically generated number series to time feature and dropped the time column as index is now time.

9. Plot a line graph of time versus carbohydrates using matplotlib library.

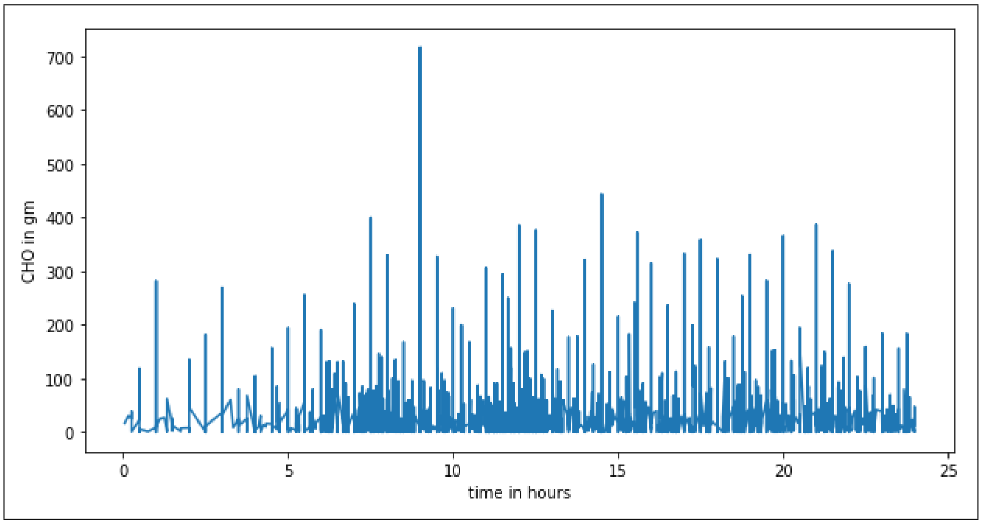

Figure 8: Graph shows time vs carbohydrates of all participants

Figure 8 shows the plot created in the steps above.

4.3 Data visualisation

The aim of this section is to try and find useful patterns(beyond time) to predict carbohydrates. After exploring and analysing data as mentioned in the section below, I decided to use the eating occasion as an important factor responsible for predicting carbohydrates.

For training the meal behaviour model, I am using data from the latest CDC's National Health and Nutrition Examination Survey (NHANES) [10], which contains information on food and nutrient intake. Some participants were asked to appear for a 24-hour dietary recall questionnaire. Interviews were taken by trained professionals for recording the diet of one day of these participants. They recorded the food items, quantities consumed and other such factors of the participants. The 24 hours recalls included both weekdays and weekend days. In this project, I have used both days for some of the solutions and only day 1 for some.

4.3.1 Relation of Carbohydrates with time

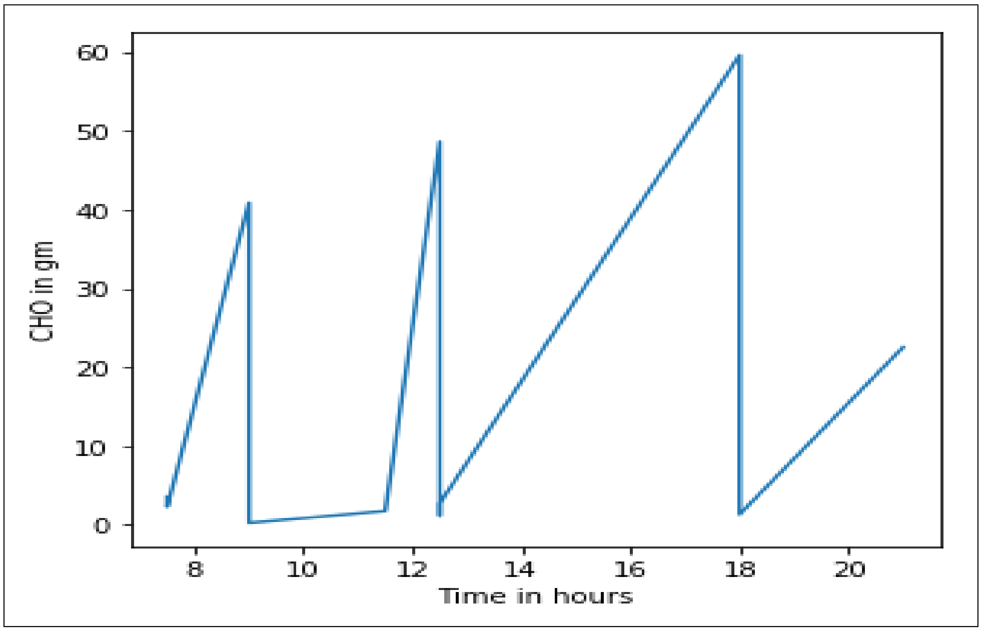

Figure 9: Time vs Carbohydrates(CHO) of one participant

I loaded the dietary interview data obtained from reference [10] in my Jupyter notebook. I created a data frame in Python Pandas with only time(DR1_020) and CHO level(DR1ICARB) of a person corresponding with the serial number(SEQN) 83732. I plotted a line graph with time(in hours) on X-axis and CHO level on y-axis. After plotting the graph, I was able to see that the carbohydrates level peaks around 9 am, 12 pm and 6 pm thus showing that carbohydrates level in our body is higher when we have breakfast, lunch and dinner. The CHO level remains almost constant between 9 am to 11 am and between 1 pm to 5 pm. Thus showing that the person did not eat anything between this period.

Carbohydrates significantly affect the insulin response of our body. There are some carbohydrates like beans, legumes and vegetables that are high in fibre and low in simple sugar. Not much insulin is required for such food. Whereas, there is some food which has refined carbohydrates which enters the body rapidly. It can elevate blood glucose levels and bad cholesterol and can lead to insulin resistance.

4.3.2 Relation of Carbohydrates with day of the week

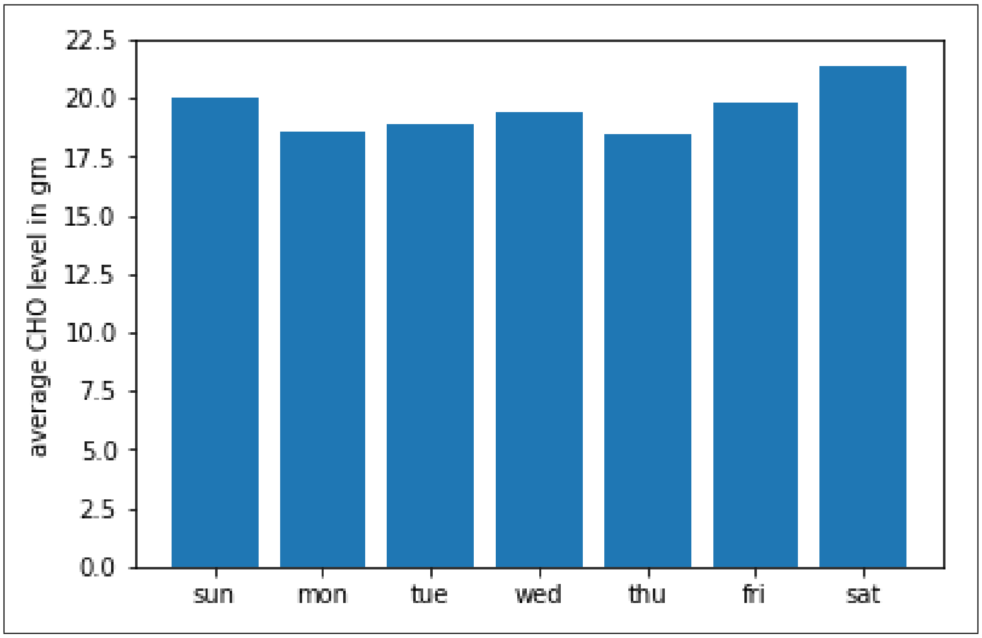

Figure 10: Bar Graph showing average CHO consumed on each day of week

I loaded the dietary interview data obtained from reference [10] in my Jupyter notebook. I created a data frame in Python Pandas with time(DR1_020), Intake day of the week(DR1DAY) and CHO level(DR1ICARB) of all people. I took an average of CHO intake of each day of the week. The codes corresponding to the day of the week are as follows:
1 - Sunday
2 - Monday
3 - Tuesday
4 - Wednesday
5 - Thursday
6 - Friday
7 - Saturday
Then, I plotted a bar graph with the day of the week and their corresponding average CHO level. After plotting the graph, I was able to see that the carbohydrates intake of people is usually more on Friday, Saturday and Sunday in comparison with the other days of the week. This suggests the

partying pattern of people because young people of age group 18 to 30 years old, often go out for parties and dinner on weekends and usually work on weekdays. Thus, their diet contains less of junk foods on weekdays, and they prefer eating homemade and healthy food. But on weekdays, as they eat in fancy restaurants, they dont restrict themselves to healthy food only and consume food which is high in carbohydrates.

4.3.3 Relation of Carbohydrates with food type

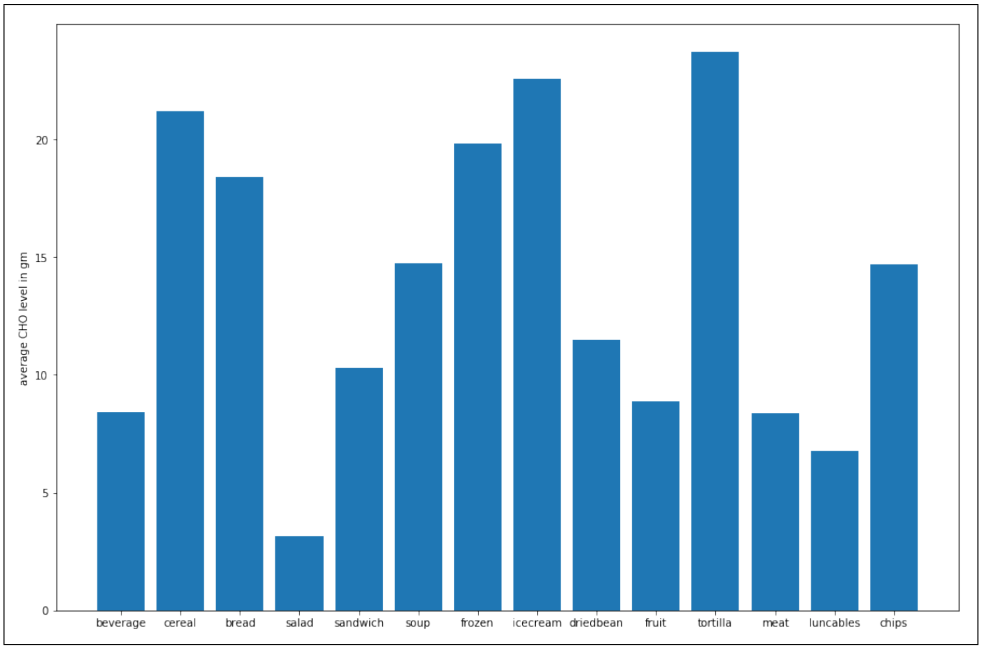

Figure 11: Bar Graph showing average CHO consumed grouped by food type

I loaded the dietary interview data obtained from reference [10] in my Jupyter notebook. I created a data frame in Python Pandas with time(DR1_020), Food type(DR1CCMTX) and CHO level(DR1ICARB) of all people. I took an average of CHO intake corresponding to each food type. The food types and their codes are as follows:
1 - Beverage
2 - Cereal
3 - Bread/baked products
4 - Salad

5 - Sandwiches
6 - Soup
7 - Frozen meals
8 - Ice cream/frozen yoghurt
9 - Dried beans and vegetable
10 - Fruit
11 - Tortilla products
12 - Meat, poultry, fish
13 - Lunchables
14 - Chips

Then, I plotted a bar graph of the food type and their corresponding average CHO level. We can see from the bar graph that tortilla, ice cream/frozen yoghurt, cereals, bread and chips are amongst the food with higher carbohydrates level. People who are trying to reduce weight try to take a less intake of such food so that the energy is obtained by the fats stored in the body, thus resulting in loss of fat for energy consumption.

According to the US Department of Agriculture [22], 100 gms of cereal has 68gm of carbohydrates, 100gm of white bread has 49gm of carbohydrates, 100gm of beer has 3.6gm of carbohydrates etc.

4.3.4 Relation of Carbohydrates with eating occasion

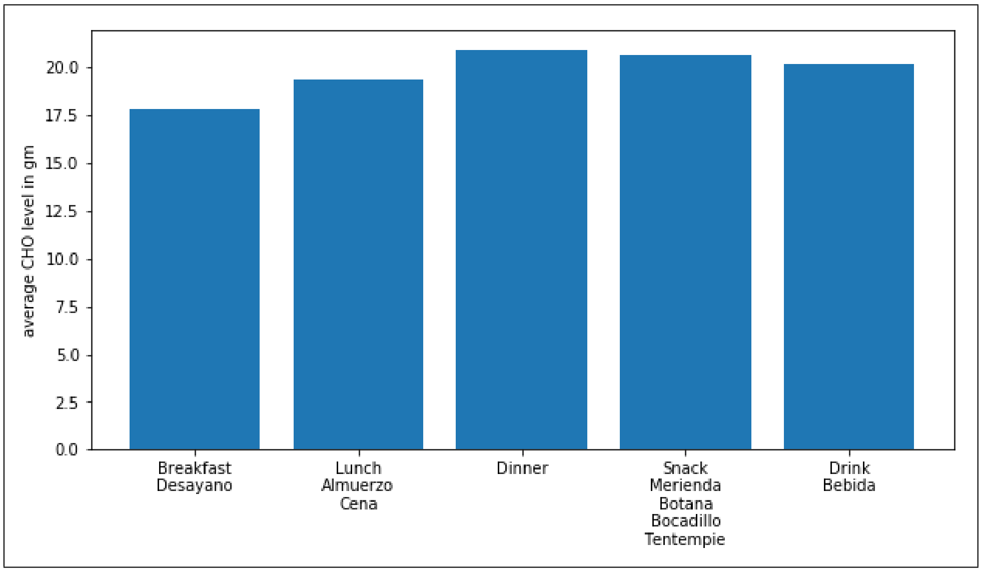

Figure 12: Bar Graph showing average CHO consumed grouped by eating occasion

I loaded the dietary interview data obtained from reference [10] in my Jupyter notebook. I created a data frame in Python Pandas with time(DR1_020), name of eating occasion(DR1_030Z) and CHO level(DR1ICARB) of all people.

As the names of eating occasion are in English and Spanish, I merged them as follows:

Merged Code	Original Code	Occasion Name
1	1	Breakfast
	10	Desayano
2	2	Lunch
	5	Brunch
	11	Almuerzo
	12	Comida
3	3	Dinner
	14	Cena
6	4	Supper
	6	Snack
	9	Extended consumption
	13	Merienda
	15	Entre Comida
	16	Botana
	17	Bocadillo
	18	Tentempie
7	7	Drink
	19	Bebida

Please note that I have ignore Infant feeding entry. Then I took an average of CHO intake corresponding to each occasion. I plotted the results on a bar graph with eating occasion on x axis and average CHO corresponding to each eating occasion on y axis.

4.3.5 Correlation matrix and graph

To predict Carbohydrates, we first need to determine how much the features are correlated to the Carbohydrates. There are numerous features in the data, 84 to be exact. We cannot use all the features as if we are also using the features which are not correlated to Carbohydrate then unnecessarily we are increasing the dimension of our model. If our model's density increases, then it will take much time to train the model, unless we are using techniques like Principal Component Analysis.

To find out the correlation between Carbohydrates and other features, I followed the following steps:

1. I calculated the correlation between carbohydrates, referred to as DR1ICARB in the data frame with all other features using the function of python pandas dataframe 'corr()'. Pandas dataframe.corr() is

a function that is used to find correlation pairwise between features of dataframes.

2. Then I filtered only those features that have their correlation value with carbohydrates greater than 0.5. I did this step to find features that are highly correlated with Carbohydrates.

3. Then I created a new pandas dataframe with features filtered out in the previous step.

4. Then I created a correlation matrix and plotted it on a colour bar to get an idea of which features are affecting carbohydrates the most. Figure 13 gives us the following correlation matrix color bar graph.

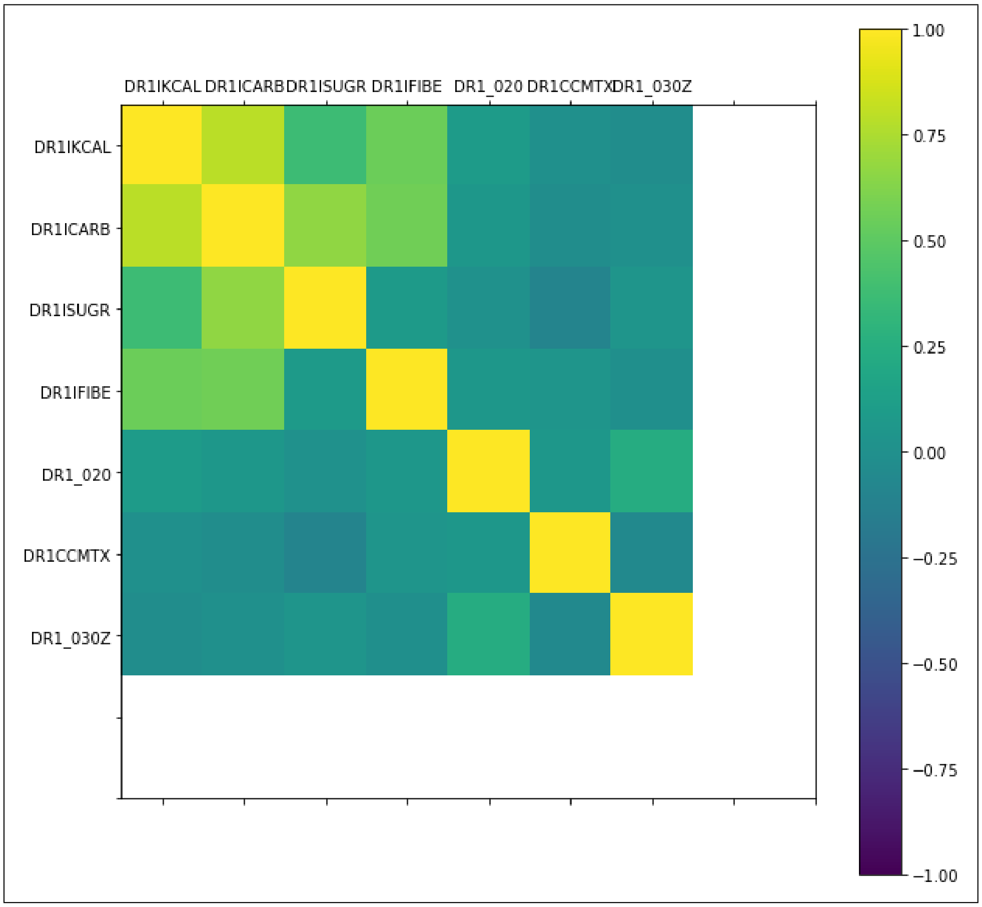

Figure 13: Correlation Matrix

From the graph, we can observe that energy, total sugar and dietary fibre are the features that are highly correlated with the carbohydrates.

4.4 Data Smoothing

A data needs to be smoothened to remove random variations in the observations and to see the structure of the causal process. Moving average can be used for data smoothing.
To take moving average of the carbohydrates, I used the rolling function on the Pandas dataframe. I created a new column and used it to store the moving averaged values of carbohydrates. Trailing moving average is used as we want to predict the next value by using the previous values. The window size I took is 500 seconds. So the moving averaged value is calculated in the following way:

$$cho(t) = \frac{1}{w}(cho(t-(w-1)) + cho(t-(w-2)) + ... + cho(t)) \qquad (15)$$

where $cho(t)$ is carbohydrate at time t and w is the window size.

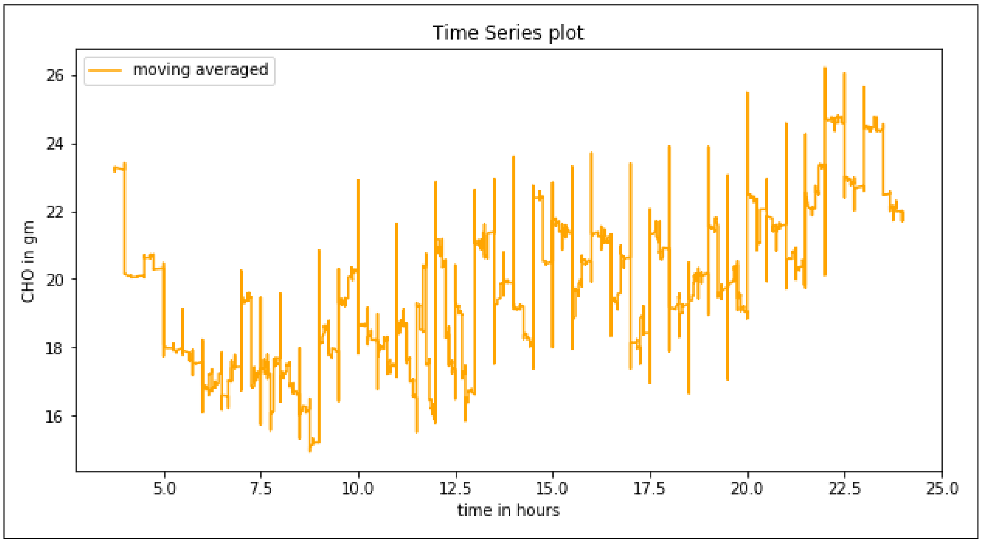

Figure 14: Time Series of all participants after data smoothing

Figure 14 shows the transformed time series.

4.5 Prediction on data

After the data is transformed, it is now appropriate to apply different models on the data and see which gives the least mean square error and the best training score.

However, for that, we need to declare the sample or independent value X and the dependent value y.
X is calculated as the values corresponding to the index of dataframe, which is the time in hours.
y is calculated as the values corresponding to the column of dataframe created as a moving average of carbohydrates, which is the carbohydrates in gm, corresponding to the time.

4.5.1 Linear Regression

As we have to predict a continuous value, my first choice of a regression model is linear regression. The dependent variable, y would be carbohydrates and the independent variable, X would be time. So, given present values of the time and previous value of time and carbohydrates, we can predict the present value of carbohydrates.

I have used scikit-learn library to apply regression models on the data. Linear regression is used in scikit-learn in the following way:

1. Fit the model using X and y.

2. Then use the model to calculate the score of X and y.

3. Use the model to predict the values corresponding to X.

4. Find mean square error between X and predicted values of y.

Following result was obtained by this model:

Regression Model	Prediction Score	Mean Square Error
Linear Regression	42.44309	2.17615

Figure 15 shows the result of this model.

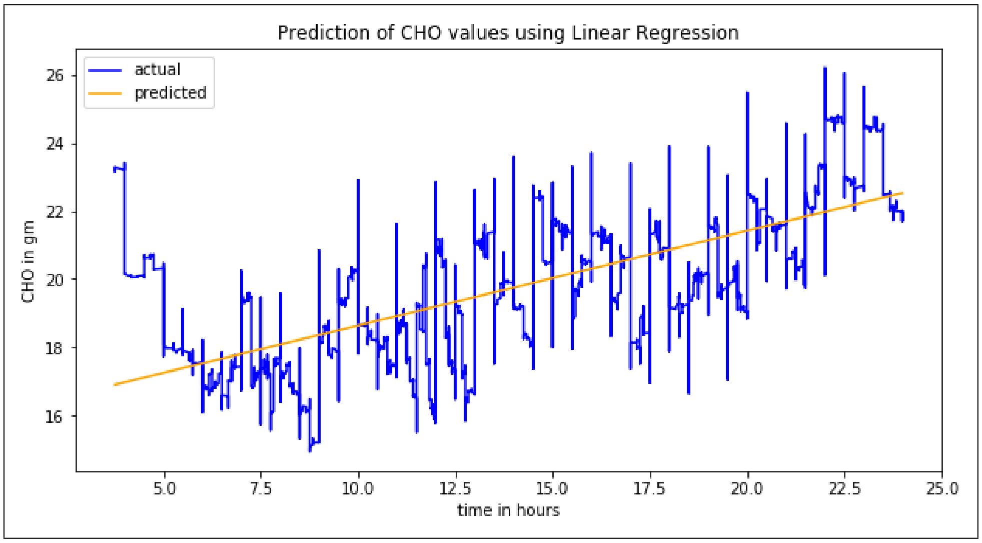

Figure 15: Linear Prediction Results

As the results are not accurate, I decided to try out other models. Also, we observe here that time alone is not sufficient for prediction. We would be needing values of past carbohydrates levels too for prediction of future values.

4.5.2 Lasso

Lasso Regression can be used as a prediction model in scikit-learn in the following way:

1. Fit the model using X and y and Lasso function of scikit-learn.
2. Then use the model to calculate the score of X and y.
3. Use the model to predict the values corresponding to X.
4. Find mean square error between X and predicted values of y.

Following result was obtained by this model:

Regression Model	Prediction Score	Mean Square Error
Lasso Regression	41.1725	2.22418

Figure 16 shows the result of this model. As the result is similar to linear regression, we discard this model. It makes sense that this model does not perform well as Lasso is used when we need to regularize our model in case of overfitting. But, we can see in the Linear Regression model that instead of overfitting, there is underfitting. Therefore there is a need to use another model.

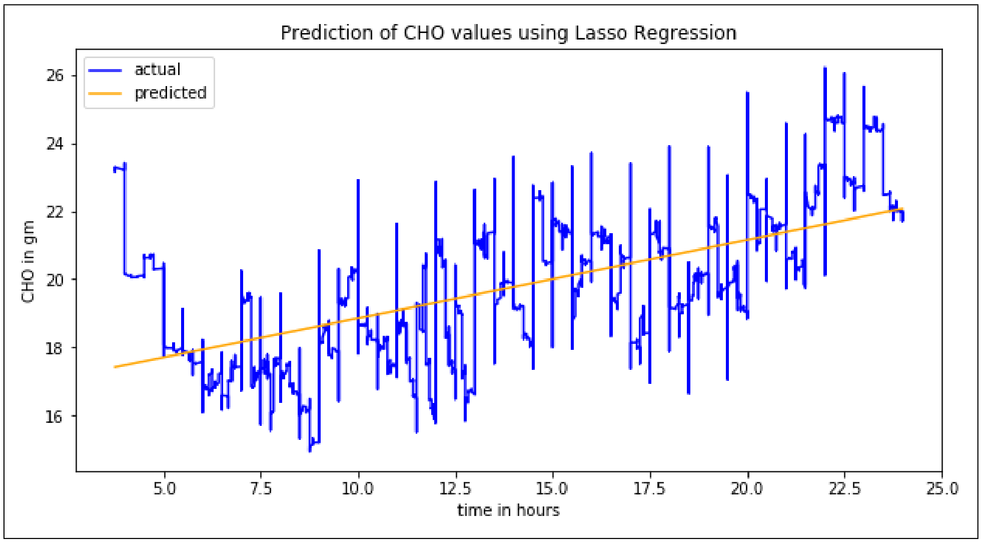

Figure 16: Lasso Regression Results

4.5.3 Ridge

Ridge Regression can be used as a prediction model in scikit-learn in the following way:

1. Fit the model using X and y and Ridge function of scikit-learn.
2. Then use the model to calculate the score of X and y.
3. Use the model to predict the values corresponding to X.
4. Find mean square error between X and predicted values of y.

Following result was obtained by this model:

Regression Model	Prediction Score	Mean Square Error
Ridge Regression	42.44308	2.17615

Figure 17 shows the result of this model.

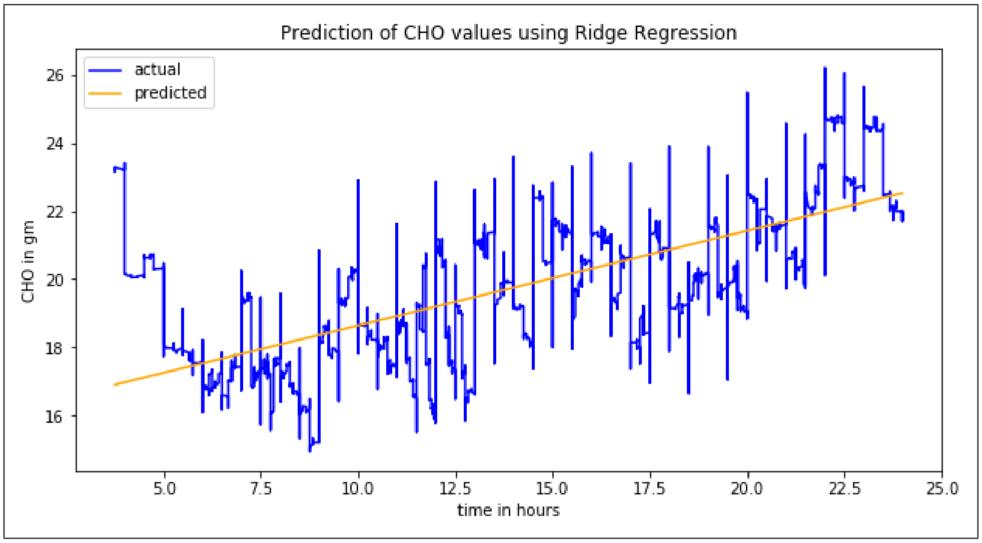

Figure 17: Ridge Regression Results

Although the results are better than Lasso, they are similar to Linear Regression and hence the accuracy is not good. Therefore, I tried random forest.

4.5.4 Random Forest

Random forest is an ensemble machine learning method which makes predictions using many regression decision trees. I have used it to predict carbohydrates in the following way:

1. Use RandomForestRegressor class of sklearn library to create a regressor with n_estimators i.e. number of trees in the forest as 1000.

2. Fit the regressor with X and y.

3. Use the model to calculate the score of X and y.

4. Use the model to predict the values corresponding to X.

5. Find mean square error between X and predicted values of y.

Following results are obtained:

Regression Model	Prediction Score	Mean Square Error
Random Forest Regression	71.36934	1.08249

Figure 18 shows the result of Random Forest Regressor. Since, this model gives the best result out of all the models I have used, I decided to go ahead with this model.

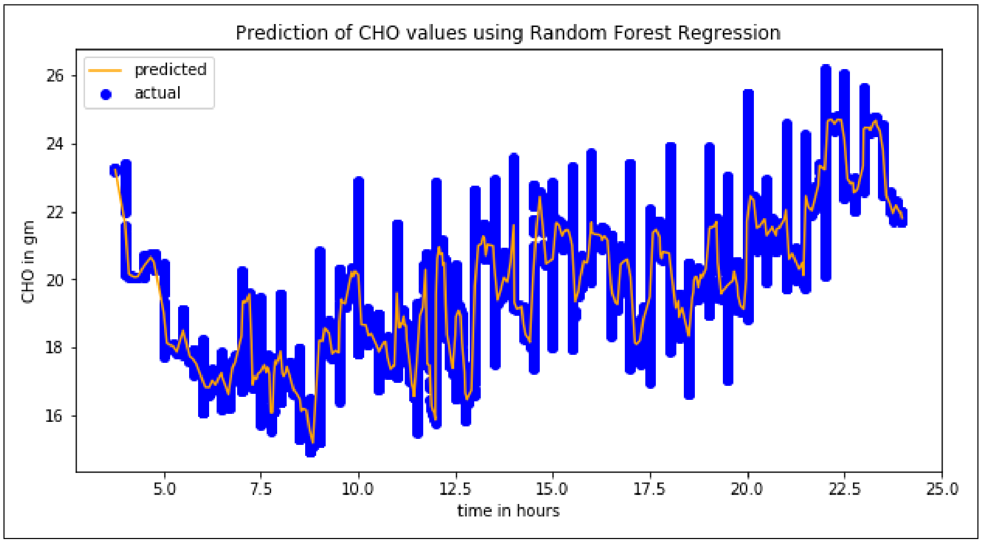

Figure 18: Random Forest Regression Results

4.6 Unsupervised Clustering

The results obtained by applying models on the whole time series can be improved if we group the data into different clusters. The idea is to execute supervised and unsupervised clustering and then observe which method gives the best result.

4.6.1 Silhouette Plots

We can choose the favourable number of clusters to create to get the best results by using Silhouette plots. We can calculate the silhouette score of each value of K(number of clusters) by using the silhouette_score class of sklearn.metrics library in the following way:

1. Create an array or list of all the possible values of K.

2. For each value of K, do the following:

   (a) Use class KMeans to create K clusters.

(b) Use the method fit_predict to compute cluster centres and predict cluster index for each sample.

(c) Compute silhouette score for each case using euclidean distances.

3. Finally choose the K which gives the maximum score.

After applying this on the data, I was able to produce the following results:

Number of clusters	Silhouette Score
3	0.4330728162294243
4	0.4399879038020235
5	0.5027783838743474
6	0.5244763371351162
7	0.539906066360256
8	0.5448062291149842
9	0.5644800952127745
10	0.5586769520342386
11	0.5359586600814948
12	0.5138428316388595

The maximum score is 0.5644800952127745 thus the value of K becomes 9.

4.6.2 K-means Clustering

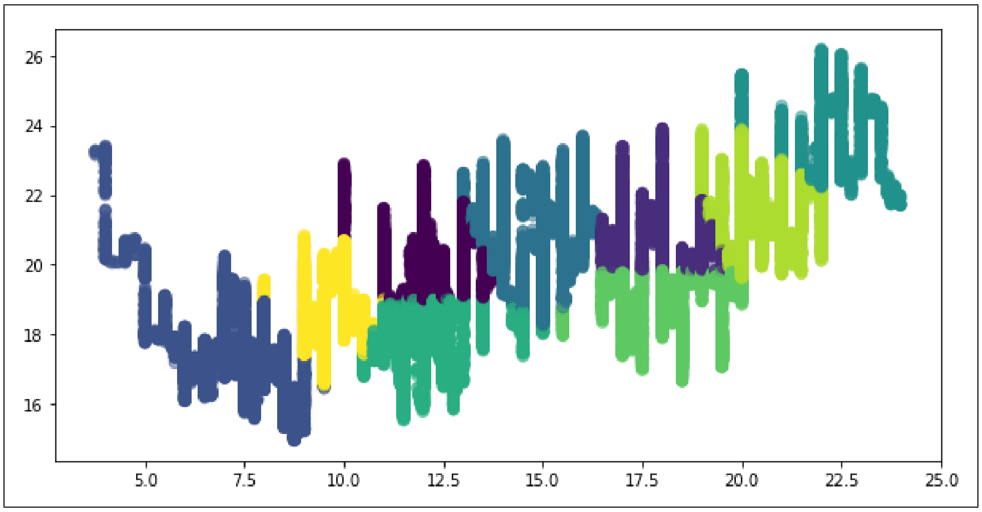

Figure 19: K means clustering with 9 clusters

I filtered out the values of time, carbohydrates and occasion of eating for clustering. I choose occasion of eating as in the exploratory analysis done of the data before, I observed that the occasion when the meal is eaten shows significant change in the carbohydrates level. After getting a suitable value of K, I created 9 clusters of the data using K-means clustering. I mapped the labels to each of the data and then grouped the data corresponding to same labels together to create 9 clusters. Then, each cluster is trained on a machine learning model.

4.6.3 Prediction on clusters

By K-means clustering, nine clusters are created on which I have applied regression. Because of the previous experiments on the whole dataset, it was discovered that Random Forest Regression was giving the best results; therefore, this method of regression is applied on each of the clusters to predict carbohydrates. The results of training each cluster with random regression are given in table below

Cluster Number	Prediction Score(%)	Mean Square Error
0	34.18252	0.51018
1	20.58946	0.46523
2	65.83207	0.41302
3	25.00793	0.86513
4	27.06841	0.73228
5	28.81698	0.44006
6	32.81781	0.42416
7	20.73889	0.59647
8	40.65534	0.48939

Figures below shows the graphs of predicted values on each cluster.

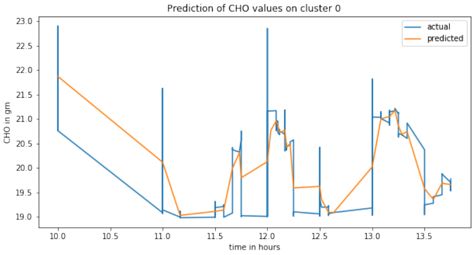

Figure 20: Regression on cluster 0

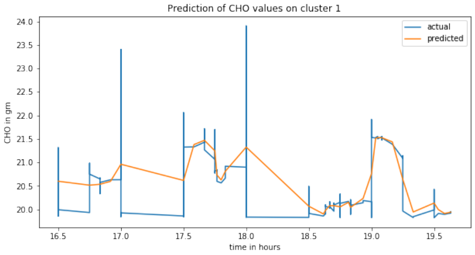

Figure 21: Regression on cluster 1

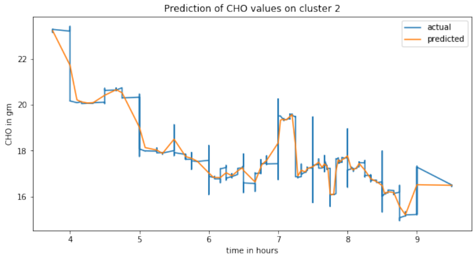

Figure 22: Regression on cluster 2

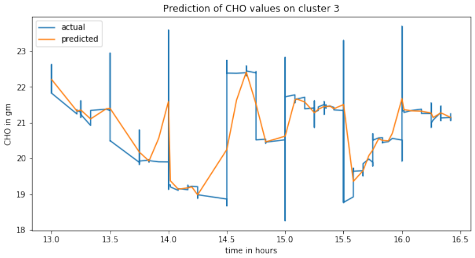

Figure 23: Regression on cluster 3

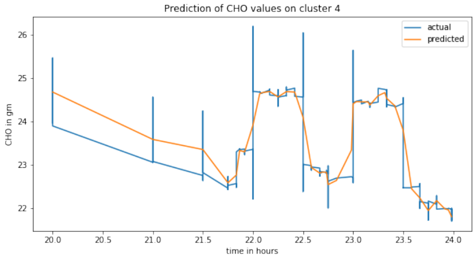

Figure 24: Regression on cluster 4

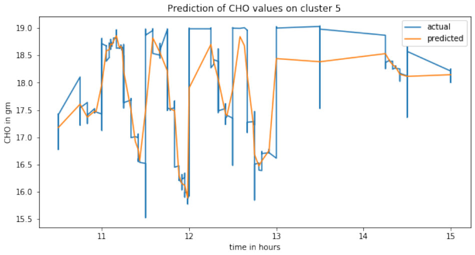

Figure 25: Regression on cluster 5

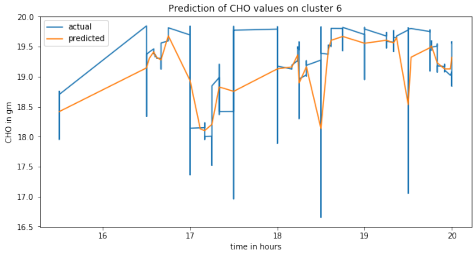

Figure 26: Regression on cluster 6

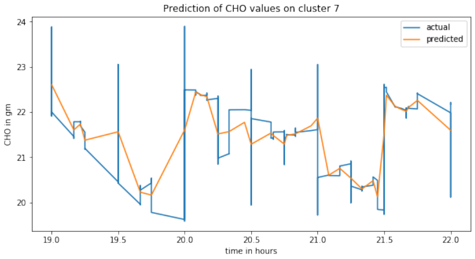

Figure 27: Regression on cluster 7

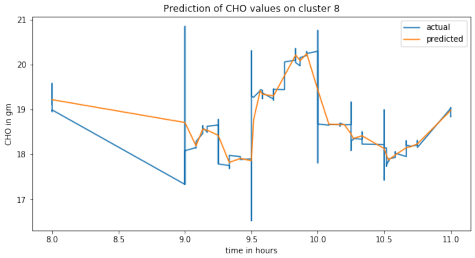

Figure 28: Regression on cluster 8

4.7 Manual Clustering

Manual clustering is done by taking into account the eating occasion of the meal. All the eating occasion are divided into five codes for breakfast, lunch, dinner, snacks and drinks. The rows corresponding to each of the five codes are clustered together to form a group. It is achieved in the code as follows:

1. Create a dictionary with codes as key and its corresponding name as its value.
2. For reach row in the dataframe, do the following:
   (a) If the occasion code is 10, then replace it as 1 for breakfast.
   (b) If the occasion code is 5, 11 or 12, replace it with 2 for lunch.
   (c) If the occasion code is 14, replace it with 3 for dinner.
   (d) If the occasion code is 4, 9, 13,14 ,15 ,16 ,17 replace it with 6 for snacks.
   (e) If the occasion code is 19, replace it with 7 for drinks.
3. Create a dataframe for cluster breakfast in which filter out and save all the rows corresponding to occasion code 1.
4. Do step 3 for lunch, dinner, snacks and drinks cluster too.

Now, perform random forest regression on each of the clusters.
The result is shown in the table below.

Cluster Number	Prediction Score(%)	Mean Square Error
breakfast	61.01900	0.77522
lunch	55.52175	1.13746
dinner	55.95793	1.20911
snacks	64.41165	1.14527
drinks	74.46232	1.10942

Figures 29-33 show these results.

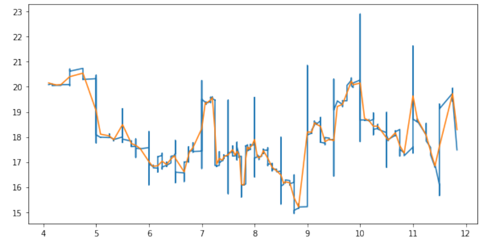

Figure 29: breakfast cluster

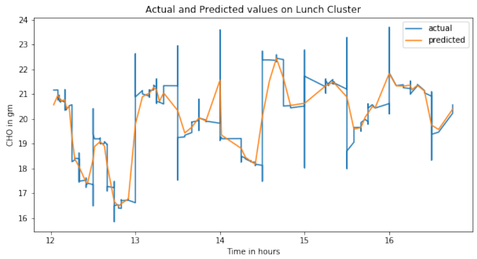

Figure 30: lunch cluster

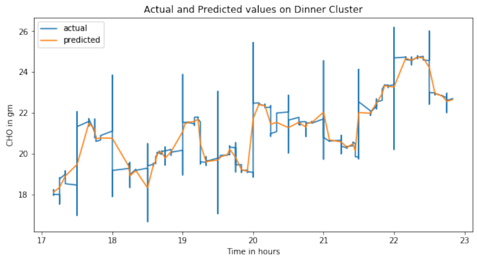

Figure 31: dinner cluster

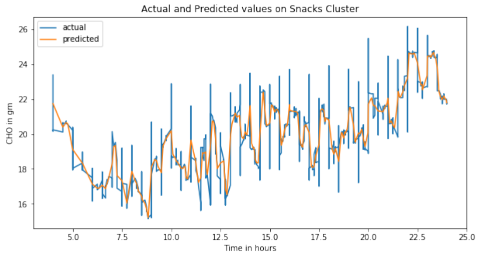

Figure 32: snacks cluster

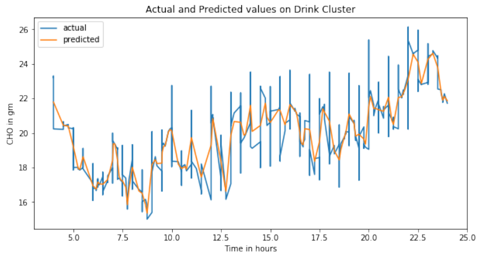

Figure 33: drinks cluster

4.8 Conclusion

4.8.1 Results

By observing the results of both ways of clustering, I observed that when I created manual clusters, I was able to predict better. In case of using machine learning in medical science, there runs a risk of diagnosis being false-negative and other such issues. To use machine learning algorithms for medical science, much testing should be done, and the accuracy of your model should be as close to 1 as possible. Therefore, I would not use my algorithm in an actual scenario as the performance is still not that good for actual use. However, if I used other models like ARIMA model or neural network, then this accuracy might have improved.

4.8.2 Future Work

Going by the article [15], it is observed that a neural network will give better results in our project. The model should be made such that a time series should be created for each patient in the dataset. A neural network is then contructed with two layers. The first layer makes short term prediction of blood glucose level by applying linear prediction. Based on the prediction, the measurement is divided into three clusters each for hypoglycemia, hyperglycemia and normal case. Then the second layer acts as a judge layer, giving the final output prediction of blood glucose.

5 Self Assessment

This project has been very informative for me as I had always wanted to know more about Diabetes and the way it affects people's meal behaviour. The way machine learning is making its way into medical science is impressive. Also, after doing this project, it helped me to understand how the knowledge that I have acquired in this course can be applied in an actual project. Personally, this also helped me in understanding more about the food I eat and motivated me to eat healthier food.

There had been many highs and lows of the project during this time. I faced some problem in understanding the time series and smoothing of it. However, once I was able to execute it successfully, I was able to move ahead quickly. Trying out different models, tuning their parameters to give the best results and then selecting the best model and their respective parameter was a lengthy and recursive process. However, because of this process, I understood in dept, the working of different regression models and my concepts are much clearer now.

About planning and executing, I would say that as I have had previous experience of working on technical projects, I had that to my benefit; therefore I was able to plan out an effective project plan. But, I felt that if I had some more time, then I could have done better research on it and could have come up with a better implementation. I wished that the time allocated for this project could have been more.

Overall, my experience with this project has been very inspiring, and I would love to explore more into it and will try to implement Neural Network and ARIMA model to give better predictions.

6 Professional Issues

Machine learning is getting more popular in the healthcare sector. Doctors and patients are now trusting the diagnosis made by machine learning algorithms more than ever now. However, there should be some ethical aspects that the stakeholders involved should take care of that threatens patient preference, safety and privacy. These ethical challenges must be identified and adopted by every stakeholder involved.

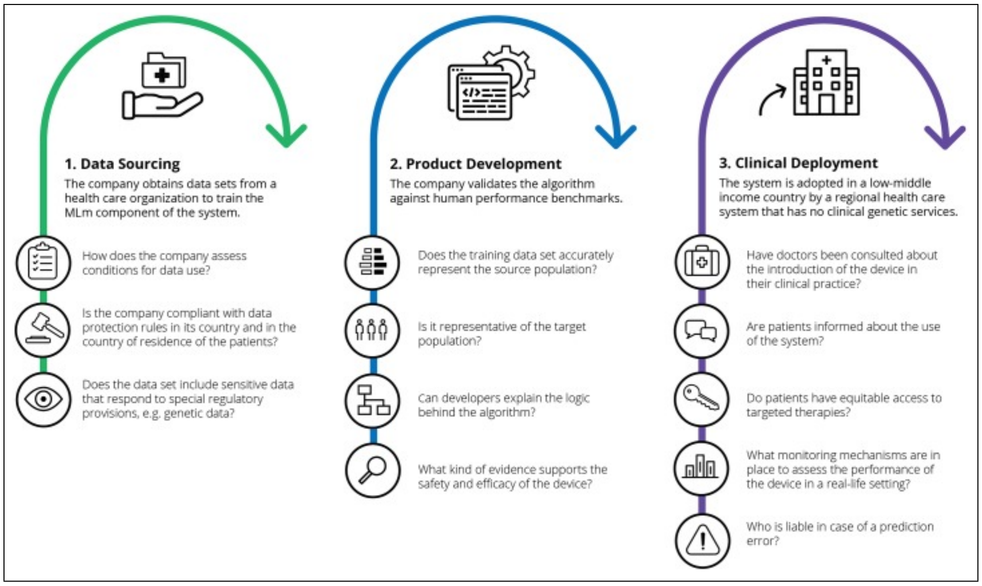
1. Data Sourcing
The company obtains data sets from a health care organization to train the MLm component of the system.
How does the company assess conditions for data use?
Is the company compliant with data protection rules in its country and in the country of residence of the patients?
Does the data set include sensitive data that respond to special regulatory provisions, e.g. genetic data?
2. Product Development
The company validates the algorithm against human performance benchmarks.
Does the training data set accurately represent the source population?
Is it representative of the target population?
Can developers explain the logic behind the algorithm?
What kind of evidence supports the safety and efficacy of the device?
3. Clinical Deployment
The system is adopted in a low-middle income country by a regional health care system that has no clinical genetic services.
Have doctors been consulted about the introduction of the device in their clinical practice?
Are patients informed about the use of the system?
Do patients have equitable access to targeted therapies?
What monitoring mechanisms are in place to assess the performance of the device in a real-life setting?
Who is liable in case of a prediction error?

Figure 34: Ethics we should consider while using machine learning in health care [23]

6.1 Patient's privacy

To train the model, the patient's data is collected, augmented and used. A study in the United Kingdom proved that 63% of the adult population is not comfortable in sharing their data for training of models and they are not in favour to replace doctors and nurses with artificial intelligence systems [17]. A recent famous case of a data breach was that of Cambridge Analytica case. The firm used personal information of millions of people, without their information, acquired through social media, to target US voters with personalised political advertisements [5]. Care should be taken when using the patient's data in the training set for building intelligent healthcare systems. Necessary permissions should be taken by the participants, and they should be informed about it properly.

6.2 Loss of doctors and nurses

The Artificial intelligence(AI) systems used in healthcare are showing significant performance and are at risk of replacing doctors and nurses. Intelligent artificial intelligence systems require just a training set for diagnosis rather than an expensive, labour-intensive and lengthy medical education required by doctors and nurses. The ethical guidelines and current medical education are lagging behind the progress machine learning systems have made in the health care field. The medical community is ill-informed of the ethical constraints associated with the uprising machine learning revolution. To take care of this, stakeholders should be encouraged to use artificial technology as a complementary tool and not as a replacement of doctors and nurses. Medical education should be re-framed, and students should be trained to interact and manage artificially intelligent machines.

6.3 Legal and health policy conflicts

There is an onset of medical malpractices and product liabilities with the rise of machine learning in health care. Black box machine learning algorithms are used by developers where they do not know inside details of the algorithm, and they cannot explain the results given by the algorithm. In case of a false negative, where a patient might have a disease, but it is not detected by the artificial intelligence system then who should be blamed in such an issue. There is an urgent need for the development of thoughtfully designed and clinically validated AI technology which can serve as a prototypical policy for the medical system.

6.4 Safety

An article states that United States decision-makers at healthcare organisations believe that AI will improve medicine by half of them think that it will produce errors and will not meet the expectations [23]. The error in results can cause a safety hazard. Suppose a patient suffering from a gruesome disease received false-negative results from an AI system, then they might lose chances of survival as they are not treated properly because of incorrect results. Care should be taken while building the model, and high-performance results should be aimed.

7 How to Use my Project

Before running the scripts, please make sure the following software and packages are either installed or accessible on the system:

1. Python3 or higher
2. matplotlib.pyplot
3. numpy
4. pandas
5. sklearn.linear_model
6. sklearn.cluster
7. sklearn.metrics
8. sklearn.ensemble

Following steps should be taken to run the scripts:

1. Download the zipped file.
2. Unzip the file. A folder with the name meal_behaviour will be created.
3. Go inside the folder; there are two subfolders inside it - one with the name report and another with the name scripts.
4. Go inside the scripts folder.
5. Download the data DR1IFF_I Data from the link **??**
6. Copy the downloaded file DR1IFF_I.XPT in the scripts folder obtained in step 4.
7. Now there are three jupyter-notebook files inside this folder.
   (a) To view exploratory analysis, open file 'visualisation and exploration of data.ipynb' in the jupyter-notebook. We can either view the results already loaded. If we want to rerun the complete file, then click on Cell and then Run All from the jupyter-notebook.

   (b) To view prediction of carbohydrates using multiple features, open file 'many features vs cho.ipynb' in the jupyter-notebook. We can either view the results already loaded. If we want to rerun the complete file, then click on Cell and then Run All from the jupyter-notebook. Please note that rerunning the whole script will take time.

   (c) To view the final implementation, open file 'meal_behaviour.ipynb' in the jupyter-notebook. We can either view the results already loaded. If we want to rerun the complete file, then click on Cell and then Run All from the jupyter-notebook. Please note that rerunning the whole script will take time.

8. Python3 py files for same are also available in the folder 'py files'. To run them do the following:

   (a) Copy data file in this folder.

   (b) Go to this folder via command line.

   (c) Then type python 'name of file.py'.

References

[1] Jason Brownlee. Linear regression for machine learning. *Machine Learning Mastery*, 2016.

[2] Jason Brownlee. Moving average smoothing for data preparation and time series forecasting in python. *Machine Learning Mastery*, 2016.

[3] Jason Brownlee. Overfitting and underfitting with machine learning algorithms. *Machine Learning Mastery*, 2016.

[4] Jason Brownlee. Supervised and unsupervised machine learning algorithms. *Machine Learning Mastery*, 2016.

[5] Carole Cadwalladr and Emma Graham-Harrison. Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach. *The Guardian*, 2018.

[6] Apoorva Dave. Regression in machine learning. *Medium.com*, 2018.

[7] Hovorka Roman Elleri Daniela, Dunger David B. Closed-loop insulin delivery for treatment of type 1 diabetes. 2011.

[8] Terry G Jr Farmer. The future of open- and closed-loop insulin delivery systems. 2008.

[9] Centers for Disease Control and Prevention. National diabetes statistics report, 2017, estimates of diabetes and its burden in the united states. *CDC*, 2017.

[10] Centers for Disease Control and Prevention. Nhanes 2013-2014 dietary data. *National Center for Health Statistics*, 2018.

[11] Institute for Quality and Efficiency in Health Care. *Hyperglycemia and hypoglycemia in type 1 diabetes.* InformedHealth.org [Internet]. Cologne, Germany, 2007.

[12] Shubham Gupta. Decision tree. *www.hackerearth.com*, 2019.

[13] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer, New York, second edition, 2009.

[14] Will Koehrsen. Random forest simple explanation. *Medium.com*, 2017.

[15] Hrushikesh N. Mhaskar, Sergei V. Pereverzyev, and Maria D. van der Walt. A deep learning approach to diabetic blood glucose prediction. *Frontiers in Applied Mathematics and Statistics*, 3:14, 2017.

[16] Nicola Paoletti, Kin Sum Liu, Scott A. Smolka, and Shan Lin. Data-driven robust control for type 1 diabetes under meal and exercise uncertainties. *CoRR*, abs/1707.02246, 2017.

[17] Michael J. Rigby. Ethical dimensions of using artificial intelligence in health care. *AMA Journal of Ethics*, 10.1001/amajethics.2019.121, 2019.

[18] Peter Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, November 1987.

[19] Elliott Saslow. Unsupervised machine learning. *Medium.com*, 2018.

[20] George Seif. A guide to decision trees for machine learning and data science. *towardsdatascience.com*, 2018.

[21] NHS UK. Diabetes. 2019.

[22] USDA. Fooddata central. *USDA*, 2019.

[23] Effy Vayena, Alessandro Blasimme, and I. Glenn Cohen. Machine learning in medicine: Addressing ethical challenges. *PLOS Medicine*, 15(11):1–4, 11 2018.

[24] Volodya Vovk. General principles of machine learning. *Royal Holloway University of London*, 2018.

[25] Volodya Vovk. Linear regression. *Royal Holloway University of London*, 2018.

[26] Volodya Vovk Zhiyuan Luo. Unsupervised learning. *Royal Holloway University of London*, 2018.