Capstone Project

Richard Biven
Machine Learning Engineer
Nanodegree
January 8, 2018

Predicting Blood Glucose Levels for Diabetics using Supervised Learning

I. Definition

Project Overview

"Supervised learning, in the context of artificial intelligence (AI) and machine learning, is a type of system in which both input and desired output data are provided. Input and output data are labelled for classification to provide a learning basis for future data processing." [15] Using my Omnipod insulin pump and Dexcom Continuous Glucose Monitor (CGM), I am starting the initial framework for creating an automated and optimized insulin delivery system (referred to as artificial pancreas and AP for the remainder of this paper). (Insert sentence that summarizes how you use supervised learning)

Diabetes Background

A Type 1 Diabetic (T1D) is an individual that has lost some or all the insulin producing beta cells of their pancreatic islets. These cells are also used to read blood glucose (BG) levels in the blood stream to inform the body if it needs to release insulin or glucagon based on the level. Therefore, diabetics must monitor their BG by drawing blood from their finger and placing that blood on a test strip for a machine to calculate the BG level. A second way is to use a Continue Glucose Monitor (CGM). This is a catheter device that is inserted under the skin in the interstitial fluid and can read a diabetics BG every 5 minutes and output it to a device or your phone. The finger sticks are still used to calibrate the CGM. The CGM is a magnificent device that helps diabetics see patterns and trends in their data.

With the data from the CGM and finger sticks, a diabetic must determine if their BG is high or low and correct their BG by decreasing the BG with insulin or increasing the BG with carbohydrates. A lot of diabetics control their insulin intake with an insulin pump. An insulin pump is catheter device that mechanically supplies insulin to the user through a set of pre-programmed settings (basal) and instant programming settings (bolus). This device is used to simulate the insulin production of the body that diabetics cannot accomplish anymore on their own. As it can be seen from my description above, life as a diabetic can feel like an eternal game of seesaw.

I am a T1D and have been since 1988. The medical field has done some amazing things to help me have a healthy life, but it is still extremely hard to maintain a healthy BG. The American Diabetes Association decided to put more funding into hardware and software that could help diabetics like myself. "Following the launch of our Artificial Pancreas Project in 2006, our first steps were to support the development of continuous glucose monitors and increasingly sophisticated insulin pumps [1]." The goal is to find a closed loop system that can measure the BG with a CGM and output the correct amount of insulin from an insulin pump to keep BG levels within the normal range. This Artificial Pancreas (AP) project is currently being worked on by some amazing groups (TypeZero, Medtronic, BigFoot Biomedical, Beta Biomedical). These companies are writing algorithms to have the CGM and insulin pumps communicate so the BG can be regulated continuous throughout the day. In September of 2016, Medtronic has produced the first ever FDA approved AP.

This work is amazing and very important to me. As an engineer for over 10 years, I am always searching for ways to advance technology. Therefore, I am writing my own AP algorithms to better understand the system as well as possibly find an algorithm that might prove useful to other AP efforts and can help diabetics in the future. It should be noted that my methods are not an attempt to duplicate other companies' work. I am making my own assumptions and calculations. If my algorithms are similar to the methods of the previously mentioned companies, it is simply because we approached the problem in similar ways.

As stated above, I am a T1D and the analysis below is designed only to address that illness. There are numerous studies on T2D, but are outside the scope of this work.

A typical for a diabetic day is like any other non-diabetic: we eat, sleep, and go about our daily routines. The big difference is, if we eat a meal full of carbohydrates and don't counteract that meal with the correct insulin dosage, our BG level will increase or decrease. Increased BG levels are called hyperglycemia, and decreased blood glucose levels are called hypoglycemia. Both situations are unsafe for diabetics. The BG data set for the analysis in this report is shown below Fig. 1:

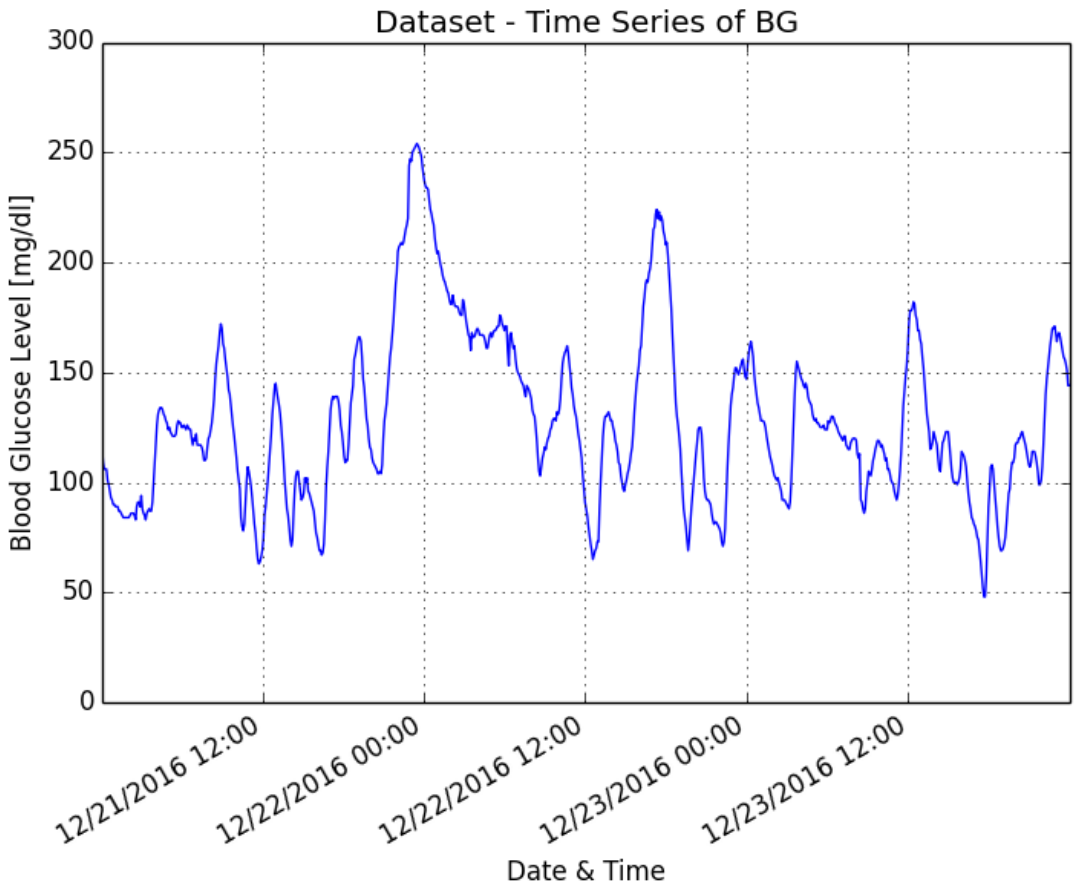


Figure 1 - BG Dataset

Hyperglycemia is considered a BG level above 140 mg/dl (milligrams of glucose per deciliter of blood) for an extended period, such as several hours. Hypoglycemia is considered levels below 70 mg/dl. BG level within these two limits are considered normal and ideal for a diabetic to maintain. Diabetics BG

levels will fall out of range after meals or during workouts, however, if the BG is return to normal, the damage caused by these states will be minimized. As seen in Fig. 1, my BG will go into the hyper and hypoglycemic ranges. It is worth describing how prolonged periods in hypo or hyperglycemia can effect a diabetic.

Hyperglycemia

Hyperglycemia is an abnormally high blood sugar. Hyperglycemia BG levels are considered above 140, but typically symptoms are only caused after significantly elevated levels, such as over 200 mg/dl. Signs of hyperglycemia vary per the individual, but typical symptoms are: frequent urination, increased thirst, headaches, fatigue, and blurred vision.

If the BG remains high, a diabetic can develop what is known as ketoacidosis, which is a build-up of ketones in the blood and urine. When a diabetic is in ketoacidosis, the symptoms are: nausea and vomiting, frequent urination, weakness, dry mouth, and sometime diabetics can fall into a coma and sometimes death. It is common to think that comas are only caused by hypoglycemia, but the reverse is true, just not as frequent.

It should be noted though hyperglycemia is undesirable, the effects take significant time for the above symptoms to take place. The remedy for hyperglycemia is to add insulin to reduce the BG. Frequent episodes of hyperglycemia can also tax other organs and contribute to heart and kidney diseases, diseases of the eyes and circulation issues that could lead to amputation.

Hypoglycemia

Hypoglycemia is an abnormally low blood sugar. The symptoms that occur in a hypoglycemia state are almost instantaneous. The body is looking for glucose to perform its normal operations and inadequate glucose is available. The following are typical signs of hypoglycemia: heart palpitations, fatigue, pale skin, shakiness, irritability, sweating, anxiety, and hunger. If the hypoglycemia worsens, the signs become: inability to complete routine tasks, blurred vision, seizures, loss of consciousness, and then possibly death.

The remedy for hypoglycemia is to add glucose to the body. This is typically done by eating, but can be achieve by medicine to stimulate glucose release in the liver.

Problem Statement

Maintaining a normal BG range is the how and why of a TID. My program is designed to evaluate several days of blood glucose levels, along with food and insulin intake, and then predict future blood glucose levels to help diabetics or machines make corrections with confidence.

As a diabetic, the BG is dependent on 7 main features:

1) Carbohydrates (*C*) – Food eaten with carbohydrates increase the BG. Carbohydrates are rapid digestions and total conversion to glucose. A carbohydrate works almost immediately and peaks at 45 minutes to an hour. The final portions of carbohydrates wear off at about 4 hours.

   A typical carbohydrate nutritional absorption looks like Fig. 2:

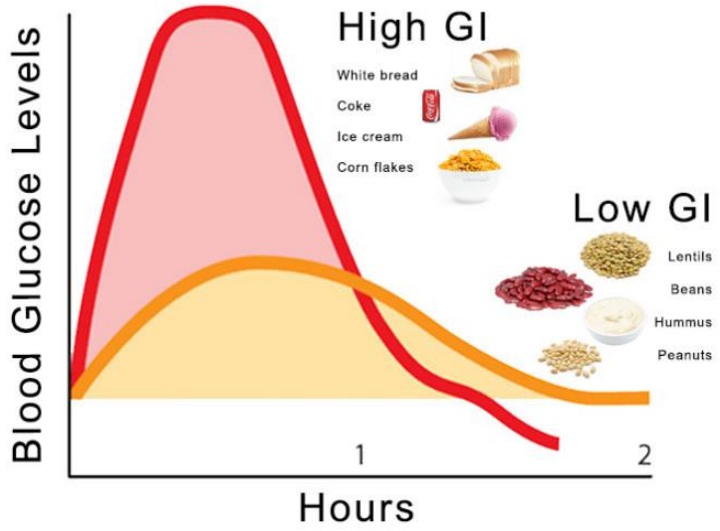


Figure 2 – Carbohydrate Nutritional Absorption

As seen in Figure 2, different types of carbohydrates absorb differently. However, for this analysis, I am assuming only High Glycemic Index (GI) absorption rates based on the foods eaten. This is a source of noise or error and will be discussed in Reflection.

2) Bolus Insulin (*B*) – Insulin used as a one-time insertion to counteract the intake of carbohydrates. "It starts working approximately 15 minutes after injection and peaks at approximately 1 hour but continues to work for two to four hours." [3]

I am a pump user, so the only type of insulin used in this analysis is rapid-acting and it looks like Fig. 3:

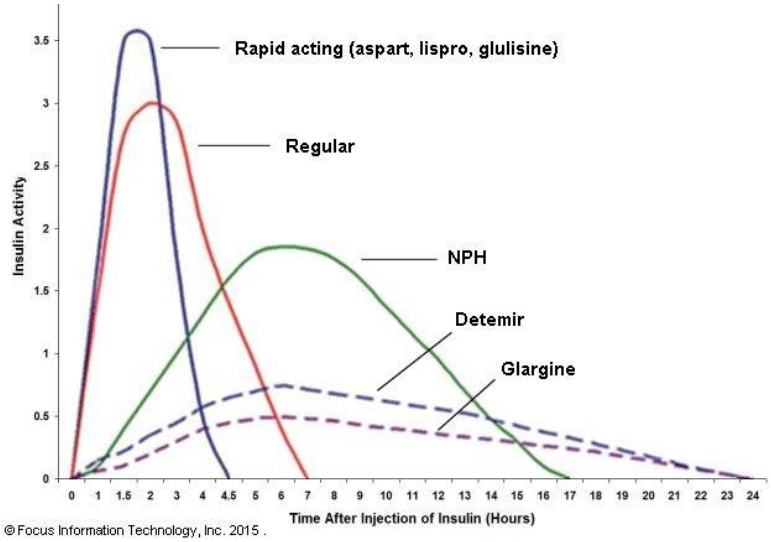


Figure 3 - Insulin Absorption Rates

I needed to create two curves that represented the carbohydrate and bolus ingestions over a period of about 4 hours. I used a Rayleigh curve for the bolus curve and a log-normal distribution for

the carbohydrates. The log-normal was used to get a little more spike in the carbs because after years of experience, the carbohydrates always reacts faster than the bolus if ingested at the same time[1]. Both curves are normalized so the area under the curve is equal to 1 and can be multiplied by the scalar value of the bolus or the carbohydrate taken. These two curves together are shown in Fig. 4:

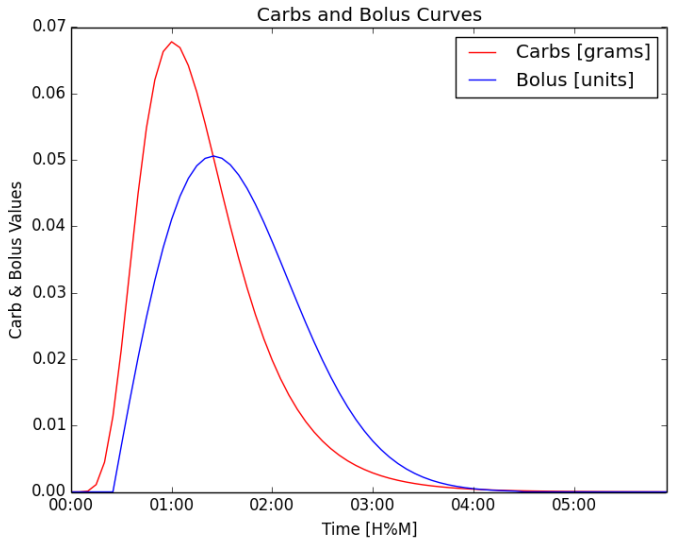


Figure 4 - Carbohydrate vs Bolus Curves

3) Basal Insulin (BI) – This is a programmed set amount of insulin that is used to counteract the glucose produced by the body to perform normal functions. The basal rate is programmed in as a step function, meaning the user programs a number of units for a number of hours. My basal rate program for this analysis is displayed in the table below.

Table 1 - Daily Basal Rates

Time	Units/hour
12:00AM	1.05
4:00AM	1.30
5:00AM	1.45
6:00AM	1.30
6:30AM	1.35
3:00PM	1.25
6:30PM	1.40
7:00PM	1.50
10:30PM	1.05

Insulin is distributed through the body over 4 hours. Therefore, the bolus curve is used to create the basal insulin for calculations. The daily basals are shown in Fig. 5:

[1] There are drugs available to decrease the peak time for insulin, meaning the insulin will act faster. However, we are not assuming this for this report.

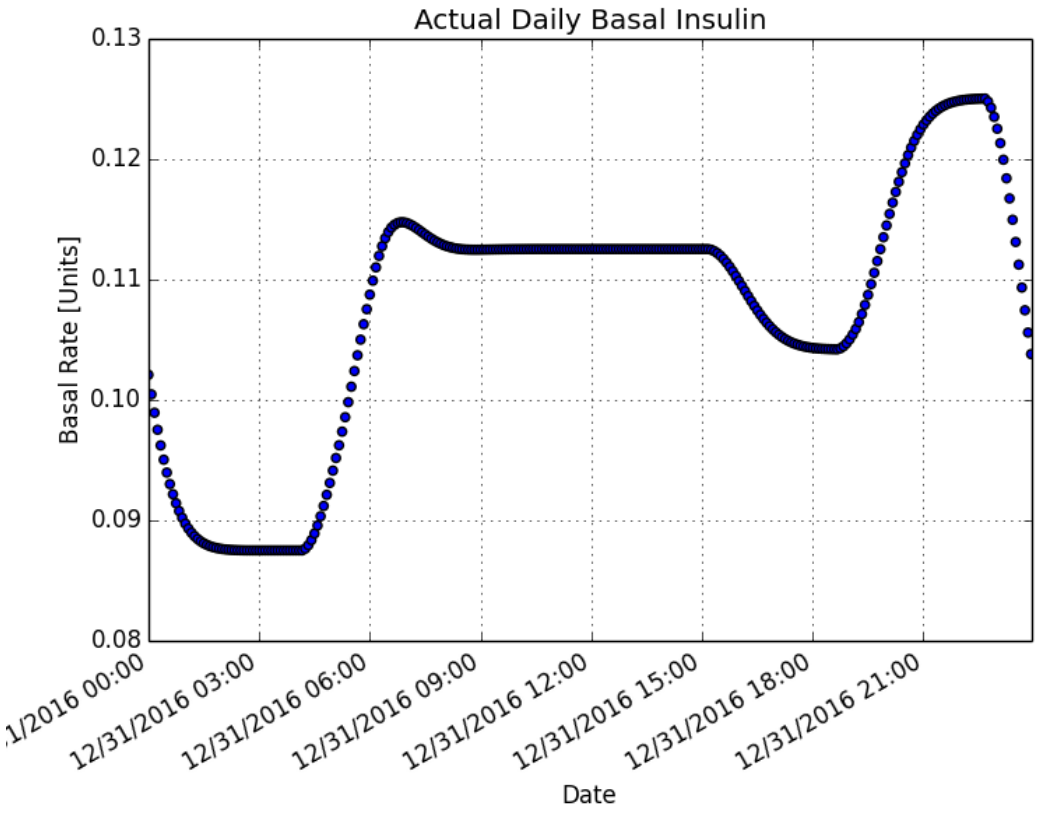


Figure 5 - Daily Basal Rate using Ingestion Curves

4) Basal Glucose (BL) – This is a continually changing feature of the body. If the body needs to send glucose to the body (brain, nerve, muscle) to perform a simple act, the liver will produce the glucose necessary to do that.
    a. For a better understanding of features (3) and (4) and how they interact, please see quote from Endocrineweb.com:

    > "Insulin is a hormone made by the pancreas that allows your body to use sugar (glucose) from carbohydrates in the food that you eat for energy or to store glucose for future use. Insulin helps keeps your blood sugar level from getting too high (hyperglycemia) or too low (hypoglycemia).
    >
    > The cells in your body need sugar for energy. However, sugar cannot go into most of your cells directly. After you eat food and your blood sugar level rises, cells in your pancreas (known as beta cells) are signaled to release insulin into your bloodstream. Insulin then attaches to and signals cells to absorb sugar from the bloodstream. Insulin is often described as a "key," which unlocks the cell to allow sugar to enter the cell and be used for energy." [3]

I am looking to solve for this variable in this paper! However, there are numerous amounts of noise and that will be addressed later in this report.

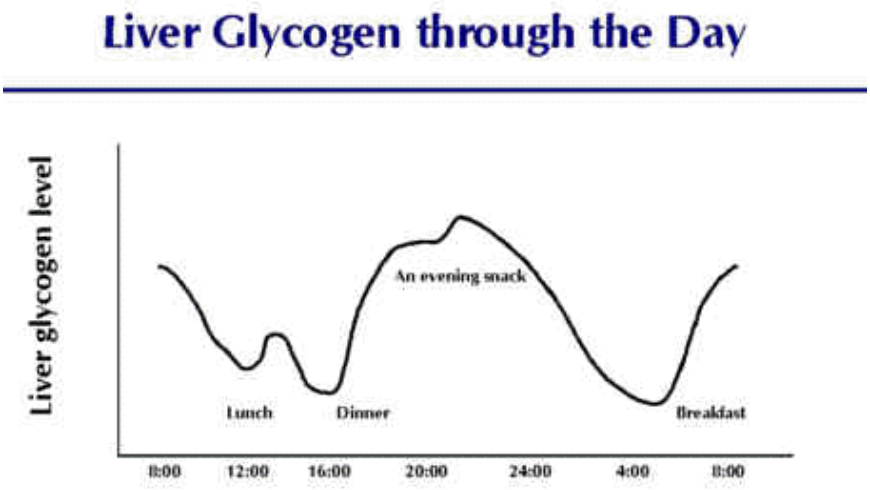


Figure 6 - Daily Basal Glucose

It should be noted that the produced regression line might not be identical to Fig. 6, but it is worth noting the characteristics. The line is not discontinuous with large jumps in values, nor does it oscillate in a short period of time. This makes sense, I do not expect anything in the body to be discontinuous.

This is the basis for **an assumption I will use throughout this report**.

Assumption 1:
"The body works on a smooth continuous line. The means no steps in the basal lines or carb and bolus ingestions. All curves will have a lower order parabolic shape. It can be assumed that a line that resemble a sine curve with a high frequency is not a normal characteristic either."

This assumption is very reasonable. Must diabetics wearing pumps will set their basal rates by fasting for a time and seeing if their BG goes up, down, or remains constant. Since basal insulin rates are programmed in at constant values, it makes sense that the body will not be jumping around if the diabetic has a constant BI rate and the BG remains constant (and it does when you get it right!).

5) Exercise – Exercise can increase or decrease the BG. Exercise will decrease the BG by using insulin and glucose more efficiently. It is possible that adrenaline released by the body for exercise can increase the BG; however, the BG is typically known to dramatically decrease when the adrenaline wears off. Exercise was not used as a feature in this analysis and is subject of future research.

6) Stress – Stress and mental state can affect the BG. If a diabetic is under significant stress for a long period, stress hormones can reduce the insulin sensitivity and increase BG. However, the diabetic data was my own and I was not under unusual stress at the time the data was collected. Therefore, stress is not used in this analysis.

7) Sickness – during periods of sickness, the body's hormones and basal rates are not in a normal state because the body is trying to fight off the sickness. These features could affect the BG;

however, I was completely healthy at the time of data collection and this feature will not be used in the analysis[2].

With these features, the BG would be calculated as:

$$BG_{t1} = BL - BI + C - B + BG_{t0} \qquad (1)$$

In equation (1), BG_{t1} is the predicted blood glucose level to come based on BG_{t0} which is the current BG. The variable *t1* is 5 minutes in the future to match the increments of the Dexcom sensor.

However, there are a few other variables that need to be quantified in order for this equation to work. These variables are completely dependent on the individual diabetic, which mean they vary from person to person. The variables are:

a. Carb Increase (*CI*) – This is the amount a single gram of carb will increase a diabetic's BG.
b. Insulin Sensitivity Factor (*ISF*) – This is the amount a single unit of insulin will decrease the user's BG.

Therefore, I need to update Eq. (1) with the added variables:

$$BG_{t1} = (BL + C) * CI - (BI + B) * ISF + BG_{t0} \qquad (2)$$

The goal of this project is to find a regression curve for the variable *BL*. In Equation (2) the only value that is not explicitly given from a device is *BL*. With *BL* generalized from machine learning, I can predict future BG based on the current state I am in, have I eaten carbohydrates or inserted bolus insulin.

I will rearrange Equation (2) to solve for *BL*:

$$BL = \frac{(BI+B)*ISF-(C*CI)+\Delta BG}{CI} \qquad (3)$$

$$\text{Where: } \Delta BG = BG_{t1} - BG_{t0}$$

I can use BL variable in the Equation (2) and predict future BG to compare to the Dexcom. However, there is a caveat with this calculation that need to be mentioned. This calculation will incorporate all noise in the system. The Dexcom is an amazing device with extremely high accuracy, but it is not perfect. Therefore, let us discuss the Dexcom briefly so we can continue down this path.

I am summarizing information provided on the company website [14]. The Dexcom G4/G5 measures it accuracy by MARD, mean absolute relative differences, for the patients referenced blood glucose measurements. A lower MARD score represents a closer or better accuracy and Dexcom is the only CGM with a single-digit MARD of 9%. This makes the Dexcom system very reliable for looking at trends in BG as well as making insulin adjustments if needed. However, there is some noise or error in the system as seen with the MARD of 9%. So, by back-calculating the basal sugars the body is producing by using the Dexcom, there is sure to be noise or error in the *BL* as well. The noise and error will be discussed at the end.

These above equations are the basis for the analysis. The method for how I used these variables features in the machine learning algorithms will be discussed below in greater detail.

[2] Features 5, 6, & 7 are very important for a AP. These will be added in future iterations of the program.

Metric

I used a typical cross-validation for this analysis; however, the score metric is not extremely high. I hypothesized this because of the dataset size and amount of noise and error. The current analysis is only using 3 days of data. As the days go up, the curves get better and better because we are using the median value of the *BL* value for that time slot. The dataset also might look like a time series analysis. This is the case for predicting, but is not necessary true for the fitting and scoring. By doing all the preprocessing analysis upfront, the dataset becomes *BL* and time. Therefore, I will still use the R2, Mean Square Error (MSE), Mean Absolute Error (MAE) values used in previous Udacity work.

These scores will not be the only metric I look at in this report. The dataset is from 12/22/2016 to 12/24/2016. The purpose of this analysis is to be able to predict BG in the future using the created *BL* values in a day. Therefore, a second dataset from 1/1/2017 2:00AM until 2:00PM was used for testing the BG with the newly acquired *BL* values, as well as the carbs and boluses from that time. The program will predict BG out one hour and score on that. Equation (2) uses BG at t = 0 as the current time to predict. I will use the Dexcom value at that time and predict out. I will use R2, Mean Square Error, and Mean Absolute Error with these predictions.

As you will see throughout this report, noise and error are my biggest issue to overcome. I think my methodology will demonstrate the basis for how my program will predict. With more training data and a few more adjustments to bolus and carbohydrate curves, the scores will only get better and better. This was seen when I initial ran the analysis with 2 days of data. Three days is better. The *BL* becomes more realistic with every extra day of data. However, there is a limit this. As discussed above, the body is always changing, which means the *BL* is always changing. Therefore, 2 weeks is the maximum for data input. More than two weeks, the body's *BL* might different from the starting *BL* two weeks earlier. Therefore, I am limiting future data sets to (at most) 2 weeks.

Exploratory Visualization

My target feature is *BL*. Fig. 7 shows a graphic of what the data looks like over a three-day period. My goal is to determine the daily *BL* for a diabetic user for prediction purposes. As shown in Equation (3), *BL* is a back calculation from the other features. Because of noise in the system, some of the *BL* values turn up negative while other turned up significantly larger than others. The assumption of this work is the body is working on a continuous smooth progression and the values will not instantaneously jump up or down, as seen in the data points. In Figure 1 - BG Dataset, the BG values follow a parametric order over time. The Dexcom system adds its own smoothing factors [4] to the data that is outside the scope of this work, but I am considering that part of the noise in the system. Since Dexcom is the most accurate CGM on the market, I am assuming this is the smallest noise I have. Noise and error will be discussed in greater detail in the Reflection section.

It is somewhat hard to see how a regression line will fit to Fig. 7. That regression line might be different for each day. It can be expected that each day's *BL* line is different. We are looking to find a generalization of that *BL* over the course of the day. Therefore, I am looking at each point in the day as a time slot to receive data. With this assumption, I took the median[3] value of the three days and used that for a new *BL* feature. By doing this, I am finding a regression line for the most likely of *BL* values.

[3] I used median instead of mean because of outliners. I can be seen that *BL* in Fig. 7 has some outliners (see spike in value around index 270).

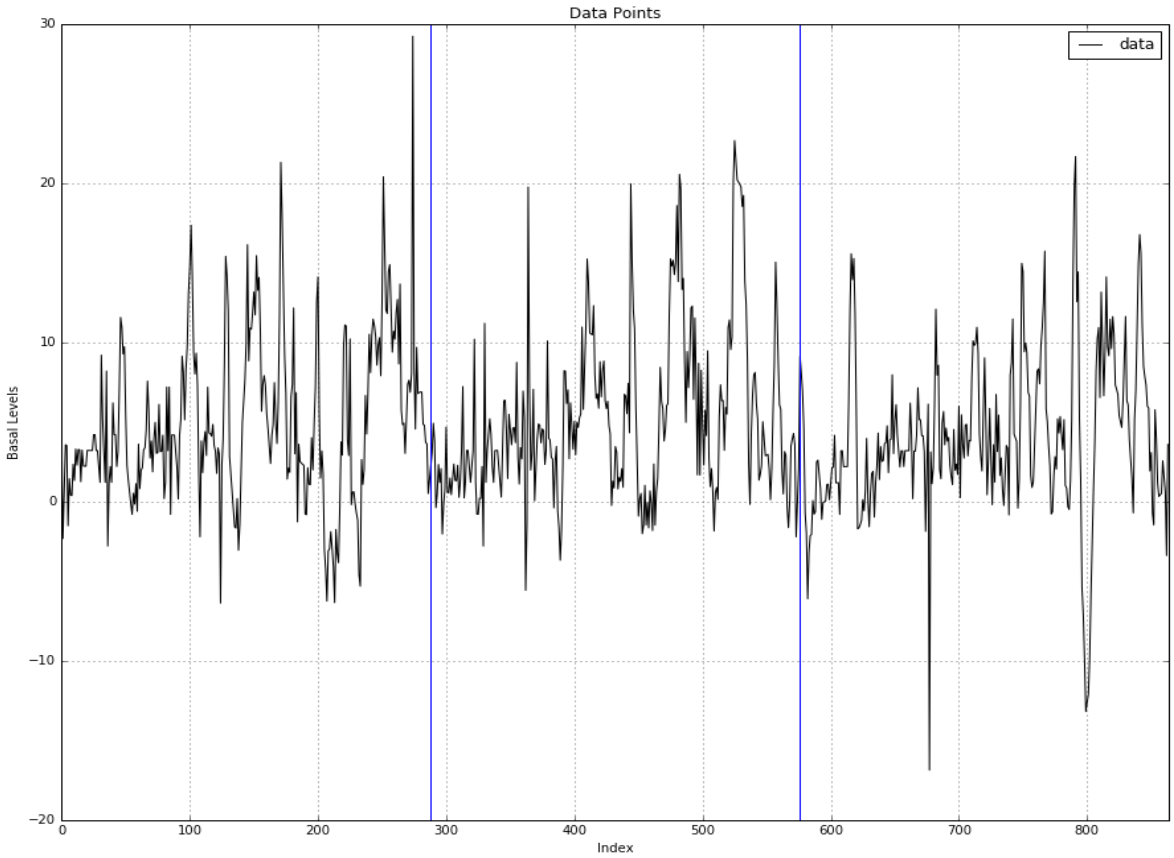


Figure 7 - Basal Glucose over a three day period

When the values of *BL* are combined using the median of the dataset, it should be noted that the features, Carbs and Bolus, are not the same every day and cannot be combined. These values are in a time series and are only used to calculate *BL*.

Algorithms and Techniques

Supervised learning can be performed as a classification or regression problem. My work is using regression because all values fall into the continuous number scale. The following features are used in the analysis:

Features:

- **Time**: Time of day (Continuous)
- **Bolus Taken**: number of units of insulin taken at a specific moment in the day (units of insulin) (Continuous);
- **Basal Insulin**: number of units of insulin delivered continuous throughout the day (units of insulin) (Continuous);
- **Basal Carbs (Liver)**: amount of glucose liver releases continuously throughout the day (grams) (Continuous);
- **Carbs Eaten**: number of carbohydrates taken at a specific moment in the day (grams) (Continuous);
- **Blood Glucose**: BG collected by Dexcom system at 5 minute intervals in the day (Continuous).

Dependent Features:

- **Bolus Ingested**: Use `Bolus Taken` to calculate how the body ingests the insulin over a period of time (Continuous)
- **Carbs Ingested**: Use `Carbs Eaten` to calculate how the body ingests the carbs over a period of time (Continuous)

The features **Carbs Eaten**, **Bolus Taken**, **Carbs Ingested**, and **Bolus Ingested** are not used in the machine learning regression. However, these features are necessary for calculation of **Basal Carbs (Liver)** and predicting future BG levels.

My future work will also include **Activity** and **Heart Rate** as features to help predict *BL* during times of exercise.

I used a few regression analyses for the machine learning in this report and the following regression methods were used:

1) Decision Tree Regression with AdaBoost - "Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features." [6]

   The DT was boosted using the AdaBoost method. "An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases." [7]

   Decision trees are simple to understand and interpret and require little data preparation. One disadvantage of decision trees is they have the tendency to create over-complex trees that do not generalize the data well.

2) Support Vector Machine (SVM) using Regression (SVR) – "A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier." [8]

   SVR is a versatile method which has different Kernel functions for the decision function and it uses a subset of training points in the decision function (the support vectors), so it is also memory efficient.

3) Gaussian Process Regression (GPR) – "Gaussian Processes (GP) are a generic supervised learning method designed to solve regression and probabilistic classification problems." [9]

   The GPR is advantageous for this work because the predictions interpolate observations and are probabilistic (Gaussian).

Using the scoring methods discussed in the Metric section, I found the following results from the regression (see Table below). Please note that I could fine-tune the regressors to completely fit the data. However, as discussed, this will not be a good representation of the body functioning under the assumption of smooth continuous actions. Therefore, the parameters were set to get a good fit to the data while remember the data is already averaged together by the time of day. I know there is noise and errors

in the data; therefore, I do not want a perfect fit. The results of the regressor fit are shown in the Implementation section.

Table 2 - Regressor Scores

Tabular Results

Regressor 1 - AdaBoostRegressor

Training Set Size	Training Time	Prediction Time (test)	R2 Score (train)	R2 Score (test)	Mean Squared Error Score (test)	Mean Absolute Error Score (test)
100	0.1050	0.0020	0.5026	0.3896	1.3210	0.8977
200	0.0290	0.0010	0.4327	0.3833	1.3348	0.9003
216	0.0570	0.0010	0.4773	0.4049	1.2880	0.8676

Regressor 2 - SVR

Training Set Size	Training Time	Prediction Time (test)	R2 Score (train)	R2 Score (test)	Mean Squared Error Score (test)	Mean Absolute Error Score (test)
100	0.0010	0.0010	0.5121	0.3605	1.3841	0.9058
200	0.0060	0.0020	0.4515	0.4180	1.2597	0.8207
216	0.0070	0.0020	0.4800	0.4122	1.2722	0.8373

Regressor 3 - Gaussian Process

Training Set Size	Training Time	Prediction Time (test)	R2 Score (train)	R2 Score (test)	Mean Squared Error Score (test)	Mean Absolute Error Score (test)
100	1.9590	0.0000	0.1318	0.1923	1.7481	0.9424
200	4.8530	0.0020	0.9597	0.7982	0.4367	0.5218
216	0.3730	0.0010	0.4841	0.4216	1.2519	0.8265

Benchmark

As discussed in Metric, a second dataset from 1/1/2017 2:00AM until 2:00PM was used for testing and benchmarking the analysis. I will use this dataset to predict the BG with the newly acquired *BL* value as well as the carbs and boluses from that day. Equation (2) uses BG at t = 0 as the current time to predict. I will use the Dexcom value at that time and predict out. I will use R-squared (R2), Mean Square Error, and Mean Absolute Error with these predictions.

For benchmarking, I am using the R2 value. "The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model. Or:

- R-squared = Explained variation / Total variation
- R-squared is always between 0 and 100%:

0% indicates that the model explains none of the variability of the response data around its mean.
100% indicates that the model explains all the variability of the response data around its mean.
In general, the higher the R-squared, the better the model fits your data." [16]

For my calculation of *BL*, my benchmark is a R2 value as close to 100%, if the line meets **Assumption 1**. Due to the noise, error, and assumptions in the analysis, I am expecting a R2 value to be lower than 100%. "In some fields, it is entirely expected that your R-squared values will be low. For example, any field that attempts to predict human behavior, such as psychology, typically has R-squared values lower than 50%. Humans are simply harder to predict than, say, physical processes." [16]

Therefore, I will use 50% for my BL calculation threshold. When I use *BL* calculations to predict future BG, I will set the R2 value to 80% because diabetics' lives are at stake.

III. Methodology

Data Preprocessing

The BG (shown in Fig. 1) does have some oscillations that seem to not fit with my assumption of smooth movements. Therefore, I have added a Savitzky-Golay filter to smooth the BG data. I want lower order curves to prevent the occurrence of a maxima and minima in succession. Because *BL* is calculated using BG, this will have a smoothing effect on the *BL* variable as well.

The Savitzky-Golay filter can be seen in the Fig 8.

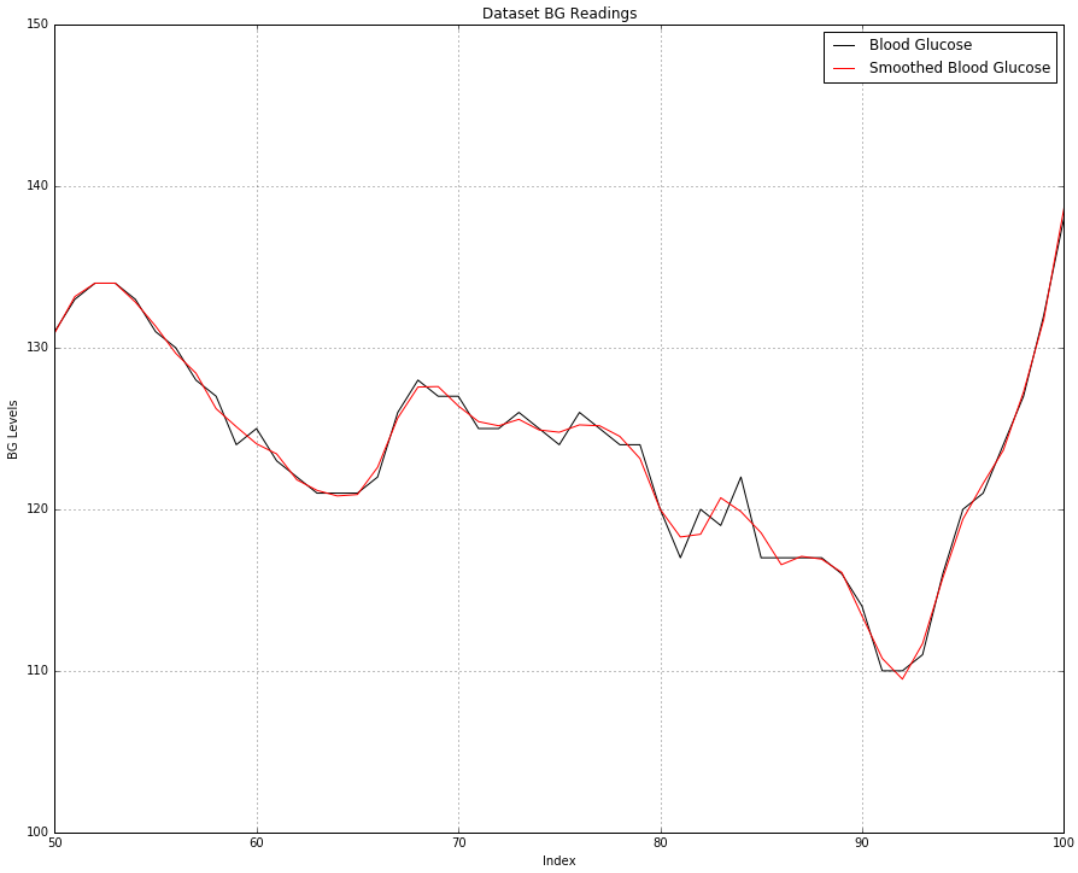


Figure 8 - BG curve with Smoothing

Implementation

The dataset consists of 864 points or three days of data at 5 minute intervals. This *BL* data was combined into one day of data which turns into 288 points.

The regressions were fit to the data using the target feature of *BL* and the remaining feature: *Time* and *BI*. As discussed above, the other features are in a time series and not adequate for this regression. *Time*, *BL*, and *BI* are all in the daily repeated form.

As seen in the Algorithms and Techniques section, the regression was performed for the three regression techniques with 100 training points, 200 training points, and 212 training points or 75% of the daily data. The other 25% was used for testing. As discussed above, the R2 scores are not above 0.5. R2 scores range from 1.0 to 0.0, where 1.0 is a perfect fit and 0.0 is no fit at all. With my scores below 0.5, I looked at a plot of the regressors over the data. Figures (9-11) show each regressor laid on top of the data. The y-axis scale is set to give a good visualization. It becomes harder to see the fit with the data going from top to bottom. Fig 12 shows the regressors on top of each other for comparison.

I did not use the out-of-the-box parameter of each regressor. They were all fine-tuned to fit the data and meet the assumption. The code used is as follows:

```
ADT = AdaBoostRegressor(DecisionTreeRegressor(max_depth=2),n_estimators=300, random_state=
0)

SVR = svm.SVR(kernel='rbf', C=1e1, gamma=0.2)

kernel = 1.0 * RBF(length_scale=50.0,length_scale_bounds=(1e-1, 1e2)) +
WhiteKernel(noise_level= 1e-5, noise_level_bounds=(1e-10, 1e+1))

GPR = GaussianProcessRegressor(kernel=kernel, alpha=0.0)
```

1) The decision tree with AdaBoost was only used with a max_depth=2. "If the maximum depth of the tree is set to `None`, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples" [6]. As discussed above, this value was kept to 2 to prevent overfitting and recognizing there is noise in the data. This parameter allowed the regression curve to not overfit the data.

2) The SVR used the standard RBF kernel, which is a Radial Basis Function kernel. The RBF is a "real-valued function whose value depends only on the distance from the origin. Sums of radial basis functions are typically used to approximate given functions. This approximation process can also be interpreted as a simple kind of neural network." [10] The parameters gamma and C were also changed in this analysis.

   The C parameter trades off misclassification of training examples against simplicity of the decision surface. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly by giving the model freedom to select more samples as support vectors

   When gamma is very small, the model is too constrained and cannot capture the complexity or "shape" of the data. The region of influence of any selected support vector would include the whole training set. The resulting model will behave similarly to a linear model with a set of hyperplanes that separate the centers of high density of any pair of two classes.

   Smooth models (lower gamma values) can be made more complex by selecting a larger number of support vectors (larger C values) hence the diagonal of good performing models" [11].

   These parameters were adjusted to meet my assumption requirement.

3) The GPR has two basic parameters to fine tune, kernel and alpha. Alpha has a default value of 1e-10 and I used a value of 0.0. The larger the value of alpha corresponds to increase noise level in the observations and reduces the potential numerical issues during fitting [9].

   I used the White Kernel for the kernel parameter. A White Kernel "is as part of a sum-kernel where it explains the noise-component of the signal. Tuning its parameter corresponds to estimating the noise-level" [12].

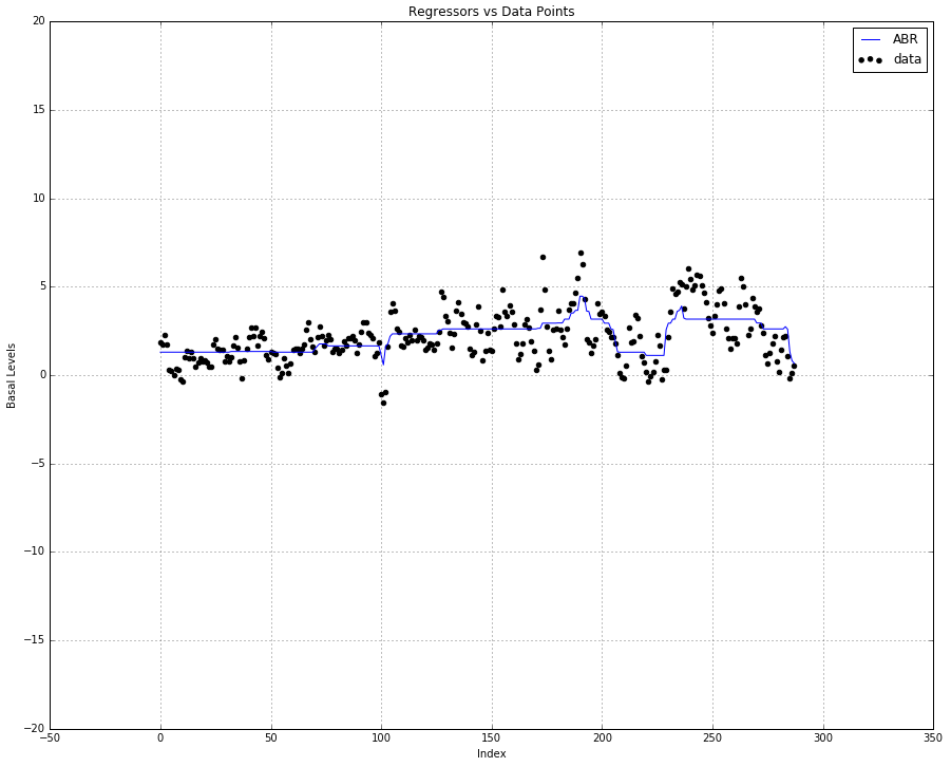


Figure 9 - Decision Tree with AdaBoost Regression Line

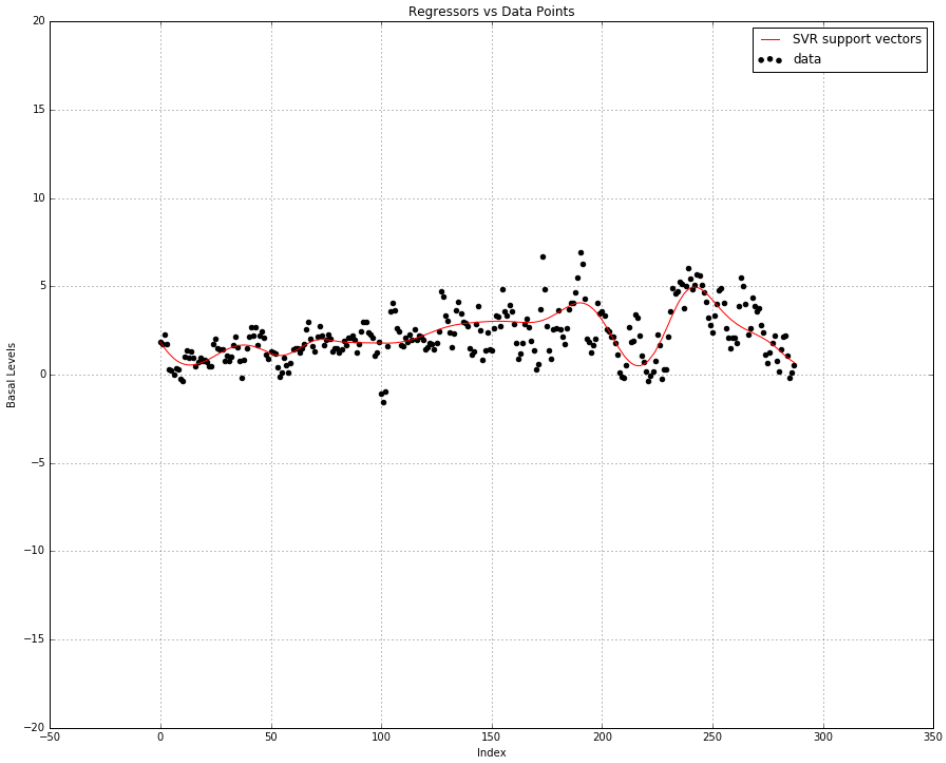


Figure 10 - Support Vector Regression Line

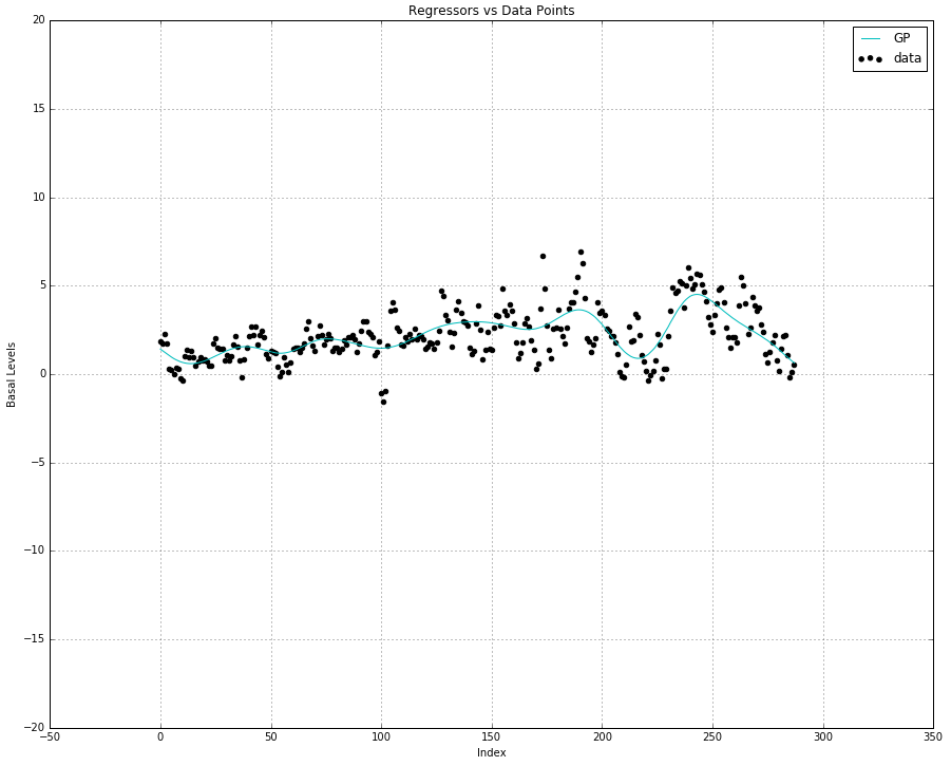


Figure 11 - Gaussian Process Regression Line

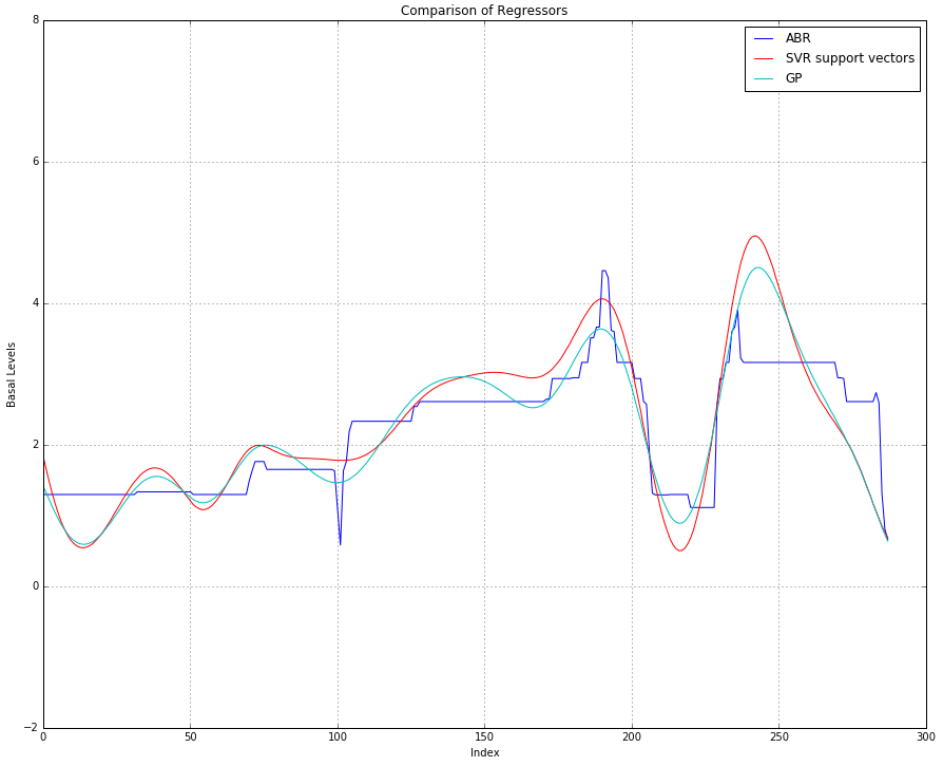


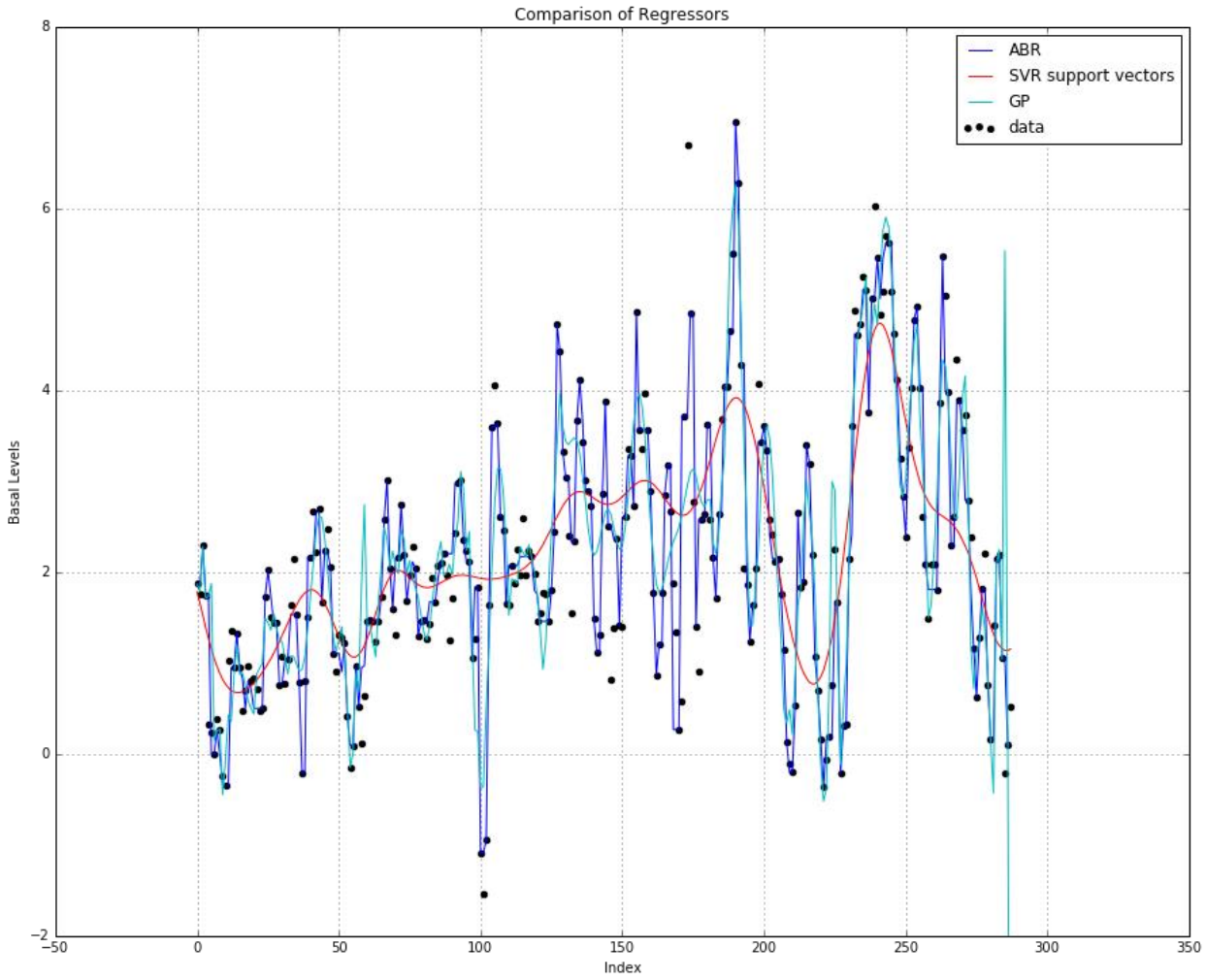
Figure 12 - All Regression Lines

By looking at each of the regressors to the data, it appears as if the regressors capture the data well. The SVR and GPR also follow my assumption by being a smooth continuous curve.

As seen in Fig. 12, all three curves follow the data in a similar manner. I chose the SVR as my choice going forward. The GPR would have been a good choice, but would drastically change with small adjustments to the parameters.

Refinement

In the Implementation section, I went through the code used for the resulting curves. If I did not make those adjustments, the regressors would of look as they do in Fig. 12:



The ABR and GPR regressors appear to overfit to the data. However, the SVR still maintains a smooth curve as I desired. This is the reason I used SVR as my regressor. I made a few adjustments to remove some of the bumps in line.

IV. Results

Model Evaluation and Validation

As discussed above, our benchmark value for the *BL* values is 50%. All three of the regression lines are very close to 50%, as seen in Table 2. I chose to use the SVR and it has a training R2 score of 48% and testing score of 41%. I believe this scores will get higher with more data and refinement to the bolus and carb curves.

In order to see if this model is viable for patient use, I need to run a new analysis with the calculated *BL* on new data to see if it can predict future BG values. Therefore, I used my data taken from 1/1/17. Fig. 13 below shows the results of predicting values 1 hour ahead of a given value from the Dexcom CGM.

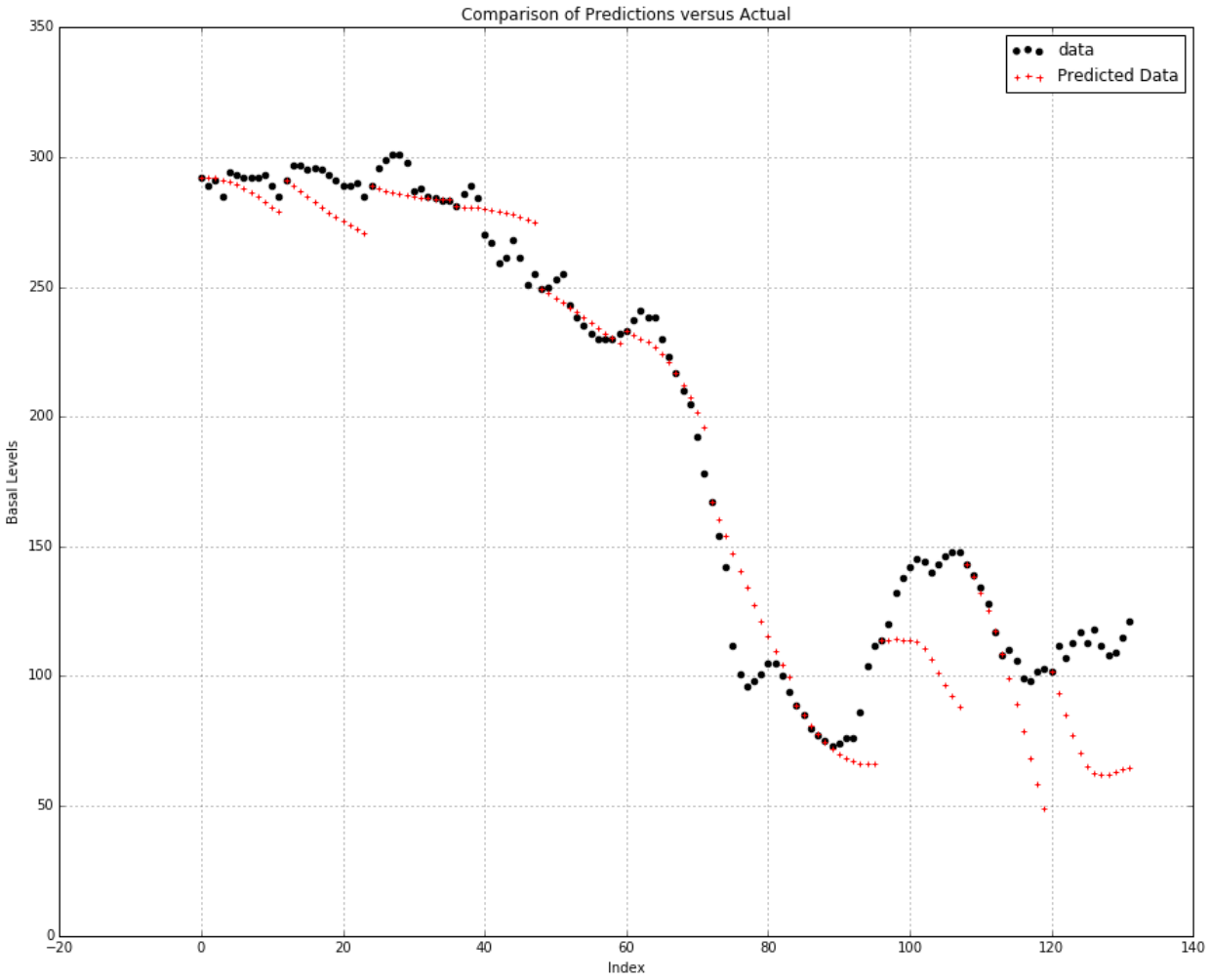


Figure 13 - Predicting BG with Calculated BL

As seen in the Fig. 13, the predicted lines are not exact to the Dexcom data. The predictions following the curve well at some of the intervals, but not in the others. This can be seen in the table below.

Table 3 - Scores from Predicting BG

Index	0	1	2	3	4	5	6	7	8	9	10
R2	-2.92	-11.9	-0.46	-0.35	**0.77**	**0.82**	0.14	-1.3	-10.9	-1.2	-72.1
MSE	32.3	167.8	70.6	203.9	20.4	67.9	492.7	338.6	1317	542.0	1842
MAE	4.9	12.2	5.9	12.3	3.4	6.4	17.2	10.6	31.7	15.1	39.6

The benchmark value for this data is 80%. Interval (4) is close to meeting this threshold & interval (5) does. In those to intervals, the prediction is acceptable for me to use in my life. However, for this program to be used all the time, I would need all intervals to be of a similar score.

I find the final model reasonable by not aligning exactly with the expected solution. I expected the variance from the Dexcom data because of the amount of noise and the fact that I have used a constant bolus and carbohydrate curve throughout the day. Carbohydrates are a very difficult parameter to model because no carbohydrate is the same. This can be seen in the Fig. 2. The same can be said for bolus

curves. As discussed above, ISF is the amount one unit of insulin changes a diabetic's BG. This value will vary throughout the day. There is a lot of variance to this number when exercise and stress are involved in the day.

It should also be mentioned that this above BG readings are a "worst-case" scenario. The Dexcom system has trouble following the BG when a large change takes place, such as going from 300 mg/dl to 70mg/dl. Therefore, I will need to run this test data with another set when the BG is not varying as much. I expect the scores to be better.

With all that said, I am happy with results so far. However, these results cannot be trusted or used for any type of BG prediction to treat Type 1 Diabetes.. **DO NOT USE THESE RESULTS FOR ANYTHING OTHER THAN ACADEMIC RESEARCH.**

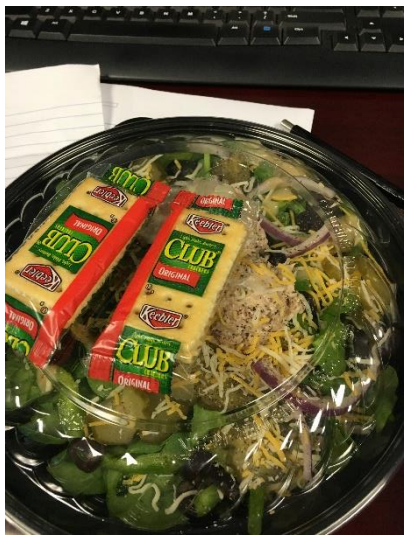
V. Conclusion

As seen above, the results are a stepping stone in the path to correctly predicting BG in the future. I will continue this work and add the addition features and functions I have discussed throughout this report. Again, if you are a diabetic, **DO NOT USE THIS WORK FOR TREATING TYPE 1 DIABETES OR ANY OTHER MEDICAL APPLICATION.**

Free-Form Visualization

The hardest part of this analysis (and the area of highest error or noise) is the ability to estimate carbohydrates that will be eaten. This is a difficult task, especially during restaurant visits or holiday parties. The images below are typical visualizations diabetics must view and determine how many carbohydrates are contained in each meal. These estimates have a large impact on BG because diabetics must bolus insulin to counteract the carbohydrates in the meals.





Reflection

The topic of noise and error has continually been mentioned in this report so I want to provide some background. One error I tried very hard to avoid is a transcription error. A user can very easily incorrectly transcribe the carbohydrate data into the program. If I ate a snack and forgot to add it to the input data, this can greatly change the *BL* variable for that time frame.

To get the value of carbohydrates from a food, I have to look at the nutritional information provided by the vendor of the food. All food is not created equal, even if it is the same product. Therefore, there is a guaranteed error or noise in the input of these values.

It should also be mentioned that proteins do have effects on BG; however, the effects are not as rapid and can take up to 8 hours for full ingestions. This is another source of error or noise in the analysis.

As discussed above, the Dexcom system is very accurate, but not perfect. Any deviation from the actual BG value is considered noise. This can be seen whenever there are larger changing in BG, such as after a high concentration of carbohydrates are eaten and the insulin is not adequately bolused. The Dexcom provides its BG value after smoothing functions and sometimes the results just aren't the same. Not shown in this report, a diabetic can take a blood reading and get 225 while the Dexcom says 195. The trend of the Dexcom is right, but it doesn't match the blood reading.

One aspect of the analysis I found interesting was the *BL* results between indices 200-250. This shows that my body produces almost no glucose and then maxes out within 4 hours. I think this might be the effect of outliners and not enough data. But the results are something I will analyze in the future.

The amount of noise affected my results in this report. As discussed above, the constant ISF, carbohydrate and bolus curves have a large impact on my results. But even with that said, I am happy with the results and I think the supervised learning technique I used is the right tool for the analysis. I feel once I add the improvements listed below into this analysis, my results will be better and hopefully viable for use.

Improvement

There are a few aspects I will improve in the future iterations of this program:

1) More Data. As discussed above, the *BL* variable will have some inherent noise associated with it. I do not think the current data set is large enough at this moment[4]. The noise can be seen when the calculation is complete and the *BL* value is negative. This is impossible and unreasonable. If I add more data, I hope the values at a specific time will begin to represent the most likely value.

2) I would like to improve in the future is the data collection process. The dataset for this work was extracted from 3 different sources:
    a) CGM – The Dexcom G5 outputs its data to one of two devices: the user's iPhone or Android phone, or the Dexcom Receiver. Neither of these devices make it easy to download the data and output it to a usable form. Therefore, to get the data, I sent the data to the Apple Health App and then exported all the Apple health information. Again, this is no easy task. The Apple Health data is exported to a .xml file which must be translated to a type that can be used in python.
    b) Omnipod – The insulin delivery system has a similar problem for download. The pump does not communicate with the user's phone so the pump needs to be plugged into a computer and the data exported. The time stamp on this data is in UNIX form which needs to be translated to the same time type used for the CGM.
    c) Carbohydrates – The carbohydrates are manually placed in a text file for the program.

[4] Please note that getting this data is no simple task. To have a complete data set, the user must measure and catalog every single carb that is eaten throughout the day. The current data set is 3 days and that was difficult. One week is the goal, but it is currently not completed at the time of this report.

3) One feature that I did not use that is commonly used in diabetic analysis is the concept of insulin sensitivity changing throughout the day. This would result in my carbohydrate and bolus curves looking different for different times of day. I think this is a very realistic feature that I should (and will) add in the future revisions. However, this will be added to the pre-processing work that will change the values of *BL*. Therefore, I should be able to use the same machine learning regression analysis above.

REFERENCE:

1) https://diatribe.org/issues/26/thinking-like-a-pancreas
2) http://www.medbio.info/horn/time%203-4/homeostasis1.htm
3) https://www.endocrineweb.com/conditions/type-1-diabetes/what-insulin
4) http://web.stanford.edu/~hastie/Papers/gam.pdf
5) https://en.wikipedia.org/wiki/Normal_distribution
6) http://scikit-learn.org/stable/modules/tree.html#tree
7) http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html
8) http://scikit-learn.org/stable/modules/svm.html#svm-regression
9) http://scikit-learn.org/stable/modules/gaussian_process.html
10) https://en.wikipedia.org/wiki/Radial_basis_function
11) http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html
12) http://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.WhiteKernel.html
13) http://denversdietdoctor.com/wp-content/uploads/2015/09/Kraft-Curves-Cummins.png
14) http://www.dexcom.com/dexcom-g4-platinum-performance
15) http://whatis.techtarget.com/definition/supervised-learning
16) http://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit