Long Term Horizon Predictions and Feature Explainability of Time Series Continuous Glucose Monitor Data

advised by Jamie Burks and Benjamin Smarr, PhD
written by Katie O'Laughlin †, Carlos Monsivias †, Leslie Joe †, Karina Kanjaria † (These four authors contributed equally to this work and share first authorship)

Abstract

Over 11% of the US population has been diagnosed with diabetes, with millions experiencing a myriad of other health complications as a result. Luckily, diabetes can be treated and managed with the proper knowledge and tools. Through the use of data science and machine learning techniques, this project seeks to help diabetes patients by analyzing what elements of their daily habits and characteristics contribute to hyper- and hypoglycemic events. Statistical and structural features are computed using close to a billion time-series glucose measurements, and applied to several machine learning models in order to understand links between glycemic events and biological rhythms. Ultimately, an XGBoost decision tree classification model is implemented for feature explainability. This model achieved an accuracy of 61.2% with hyperparameter tuning. With such a model and its accompanying front-end application, patients and healthcare professionals are able to see which features most impacted the model's predictions. This grants users the abilities to assess, understand, and potentially take actionable steps to improve their health.

Introduction

According to the World Health Organization, in 2014, 8.5% of adults aged 18 years and older had diabetes. In 2019, diabetes and kidney disease due to diabetes caused an estimated 2 million deaths. Between 2000 and 2019, there was a 3% increase in age-standardized mortality rates from diabetes. In lower-middle-income countries, the mortality rate due to diabetes increased 13% ("Diabetes").

Complications from diabetes can cause disabling and life-threatening health complications. These include, but are not limited to, damage to organ systems, retinopathy, nephropathy, neuropathy, diabetic foot disease, a 2 to 4-fold increased risk of cardiovascular diseases, and death in the most severe cases (Goyal 2023).

These statistics point to the importance of diabetes management, as diabetes can be treated and its consequences avoided if managed properly. Most medical advice related to diabetes treatment and insulin administration is informed by static models dependent on meal times and quality. Historically, the main factor used to evaluate diabetes levels is HbA1c levels (Milson 2020). The hemoglobin A1c level is the measure of average blood sugar over the past 2 to 3 months. Other factors, such as Time in Range, have also been introduced to help inform diabetes management. This project explores how to deliver knowledge that can inform diabetes treatment and insulin administration by extracting structural rhythms and other characteristics in addition to these traditional methods.

If managed properly and predicted heavily in advance, patients and caretakers would be able to decrease HbA1c variation. This would also decrease the risk of complications or life-threatening episodes. Diabetes management would become easier if patients knew when to

preemptively take insulin, pack snacks for a long trip appropriately, and surround themselves with a support system if a dangerous episode is predicted well in advance and more accurately. However, current research in the field includes these types of models dependent on traditional time in range characteristics. These prediction models do not provide much insight into what leads to the predicted glycemic events, however. There exists a gap in user understanding of underlying factors in glucose fluctuations outside of food consumption and insulin administration.

This leads to the questions addressed through this project. *Is it possible to extract labeled features from time series data that attribute the most variation to glucose levels? What, if any, dynamic or nonlinear measures could complement traditional clinical measures of glycemic variability in the assessment of diabetes control? What machine learning models can be developed with explainability included to generate predictions of future hyper- and hypoglycemic events?* This project and model addresses these questions and pinpoints the important underlying features.

Team Roles and Responsibilities

I. **Carlos Monsivias**

Solution architect, methods expert, scalability and operations expert, machine learning engineer, data scientist.

II. **Karina Kanjaria**

Project manager, scalability and operations expert, domain researcher, solution architect, methods expert, data scientist.

III. **Kate O'Laughlin**

Data engineer, visualization and dashboard developer, software engineer, domain researcher, data scientist.

IV. **Leslie Joe**

Communications manager, bookkeeper, data scientist, methods researcher, visualization expert, data scientist.

Data Acquisition

I. **Data Source**

This project uses one source of data collected from Dexcom's continuous glucose monitors. This CGM data from patients has been deidentified by Dexcom prior to transfer to the team working on this project. Glucose values have been recorded at 5 minute intervals for up to one year for each patient and this data was provided in csv files per day. Initially 10,000 patients had been provided as a part of this dataset, however due to incomplete or largely missing data, this was narrowed down to 8,000 patients. All the data was provided at once so no data ingestion stream was necessary.

II. **Data Collection**

Dexcom provided 500 gigabytes of patient data in csv files to data administrators at UCSD. This data was uploaded to the UCSD cloud based platform, Nautilus, with a persistent volume claim. The data was accessed via Nautilus and this persistent volume claim mounted on Jupyter Hub.

III. **Data Pipeline**

The data pipeline relies on PySpark to handle the large volume of data provided. It is first loaded from the CSV files into PySpark dataframes, then cleaned, repartitioned by patient, and output into a location on Nautilus as parquet files. Parquet files made it possible to quickly read and write spark dataframes with the large volume, along with future users these same flexibilities. The data was split into train and test sets and the test set was then loaded into an S3 bucket for integration with the front-end application. The application accesses the test data in the S3 bucket using private credentials.

Data Preparation

Dexcom provided data for 8,000 patients which were provided in files split by date for the 365 days between February 1, 2022 to January 31, 2023, though some patient's data started before February 1, 2022, and more started after (such as starting mid-March). When performing the initial data analysis the team found many quality issues with this dataset. Firstly, not each of the patients contained data for each day. For some, many months of data was missing. For patients which did not present with a significant amount of data, greater than 60%, they were dropped from the modeling and analysis. Such large quantities of missing data would lead to issues with the pipeline with interpolating data and modeling on too much "engineered" data. Secondly, many of the dates found in each of the separate CSV files were incorrect. The data was occasionally random in which file it was stored in, and therefore presented issues with the time-series aspect of the analysis steps. Some computationally heavy sort functions were performed on the dataset early on in the pipeline to combat this. By the nature of human-interfacing technology, some recordings were lost and those points in time were recorded as null or 0 (a glucose level that is not physically possible for a living human). The missing data could be due to their continuous glucose monitors glitching, running out of battery, restarting, or being removed. This missing data was interpolation using averages.

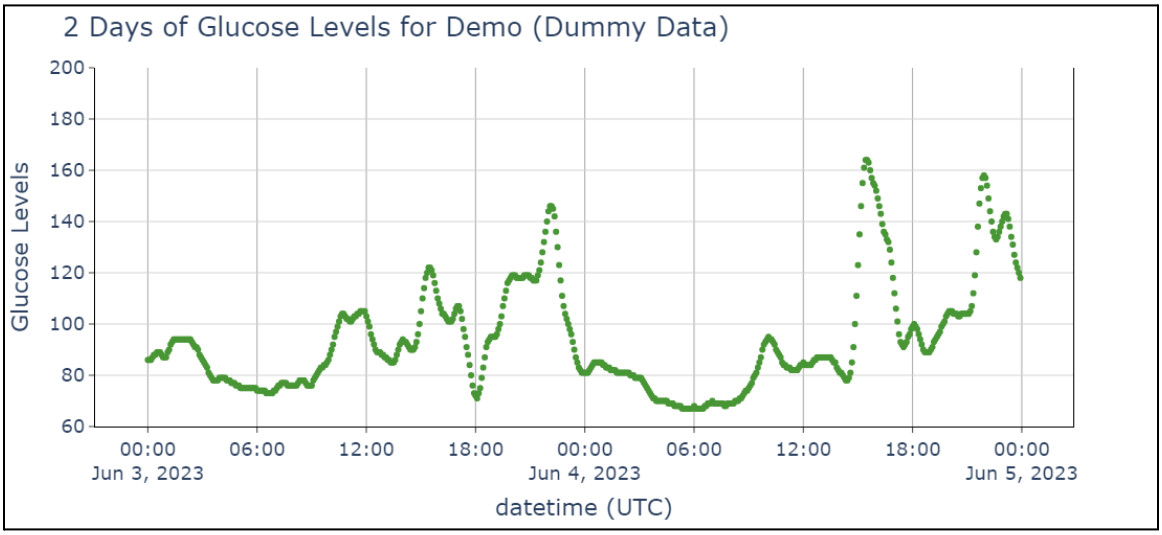

Fig. 1: Two days of a patient time series CGM data, graphed in Plotly, using an open source CGM dataset (Martin 2021) for visualization purposes only.

In order to transform the raw CSV data into usable data for the project, all of the above steps were performed by loading into PySpark dataframes, cleaning, interpolating, and saving to parquet files. These pre-processing methods helped in removing patients with large amounts of missing data so that the models were not generated on largely engineered data. They also helped to ensure that the time-series model would remain intact, a worry that arose due to the interpolation methods of inserting small amounts of missing data to avoid breaks in time. With these clean up steps, future users will be able to avoid computationally heavy pre-processing, as these sets have been saved and labeled.

After performing this preprocessing, different types of features were generated from the dataset. Many statistical features were created along with some additional structural features. These features were selected and managed by analyzing the models built from them and seeing if these features were significantly impacting performance. Data sets with the different features were also saved in parquet files within Nautilus cloud.

Analysis Methods

The chosen data analysis methods in this project were driven primarily by the data's own characteristics. Initially, methods were identified in order to perform preliminary analysis on the small sample set of data available to us of three patients. After learning the characteristics of this small set, the methods were revised and tuned for parallel processing in PySpark after calculating the estimated full dataset size and processing power needed.

As this was time series data, well-known time series analysis techniques were selected for analysis. This began with simple statistical features such as mean, median, maximum, and minimum of the given data. This data was visualized with plots over time for each of the initial three patients to understand the shape of the small sets. This was expanded to more complex calculations such as velocity, acceleration, entropy, poincare, and multifractal detrended fluctuation analysis. From these calculations in the preliminary analysis with Pandas dataframes, methods were expanded to user-defined functions in PySpark.

Once obtaining the full dataset, the volume of the data drove the decision making process on which features were included. Some calculations such as multifractal detrended fluctuation analysis were not easily compatible to scale with the PySpark data engine. For these types of calculations, analysis was performed but only selected as features after completing modeling at a later stage. For that reason, the methods of performing analysis on the data was influenced by the data characteristics themselves, which led to a strong definition of the questions addressed in the project. After observing which types of analysis and features could be pinpointed, it decided that the methods provide clarity through feature extraction for patients and providers, instead of building a deep learning model in a saturated glucose predictions field. This would set the project apart from others in the same domain.

With this knowledge and outline generated from preliminary data analysis, the final workflow was set up. It contains saved data checkpoints due to computationally expensive calculations with a lack of RAM with plenty of storage space in the cloud environment. These analyses are performed on Nautilus cloud with a minimum of 2 GPU, 4 CPU, and 128 GB memory with multiple python packages and PySpark. Team members would save data checkpoints in parquet files which are lightweight and do not require much space, instead of performing each step redundantly every time analysis or modeling was required because these

processes take large amounts of time to complete with the full 500GB dataset. The workflow is as follows.

1. Load data from csv files into PySpark dataframes.
2. Clean up data and drop unnecessary information.
3. Rearrange data by patient and time, split into training and testing sets, and save to parquet files as the first checkpoint.
4. Interpolate missing data per patient using the daily average methodology and save to parquet files as the second checkpoint.
5. Create data chunks for the time frame required for use in calculations which were extracted as features to feed into the ML models; numerical values were used to section off these chunks which resulted in grouped data per single day per patient.
6. Calculate complex structural features per patient per chunk- sample entropy, permutation entropy, short term deviation poincare, long term deviation poincare, short to long term ratio poincare, and save to parquet files as the third checkpoint.
7. Calculate summary statistics features and target values per patient per chunk- mean, median, standard deviation, maximum, minimum, average first difference, average second difference, standard deviation first difference, standard deviation second difference, count of time above range, count of time below range, total time out of range, and change in time out of range from previous chunk target variable, and save to parquet files as the fourth checkpoint.
8. Create categorical features per patient- age range groups, sex, and treatment type groups and save to parquets.
9. Merge all the generated feature data frames together to create the final data frame and save to parquets as the final checkpoint.
10. Scale the features using standard scaler and use one hot encoding on the categorical variables.
11. Create lagged values necessary for time series modeling.
12. Create an XGBoost model using the training data.
13. Test with the test-dataset and provide results.
14. Connect the model and data to the front-end user interface application.
15. Deliver to the customer.

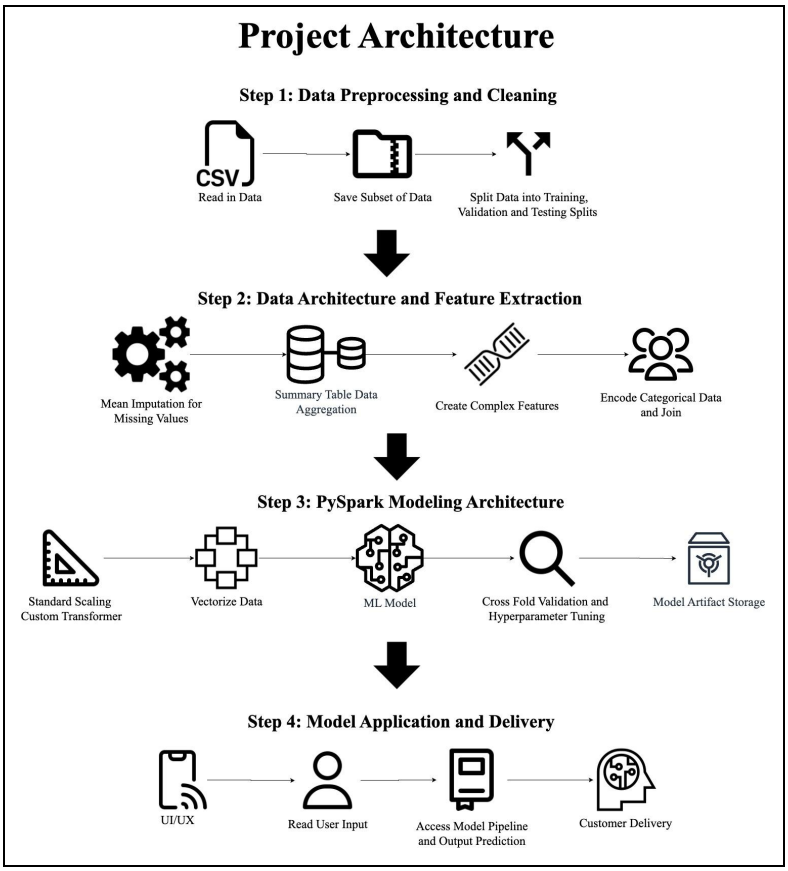

Fig. 2: The basic overview of this project's pipeline (using open-source icons through draw.io).

Findings and Reporting

Through data analytics and modeling, key findings were determined about glucose fluctuations in diabetic individuals. With XGBoost Decision Tree modeling, the features with the most impact in glucose predictions were identified for patients in future days given past data. These impactful features changed based on the type of model used and hyperparameter tuning. However, one feature that consistently presented as the most significant across most models was the scaled total out of range value for the individual. Upon looking at the data further, and ensuring that train and test sets were well balanced for glucose levels and other characteristics, it was discovered that many individuals were present with the same underlying patterns on a day to day basis. This means, for example, that patients who present out of range of normal glucose values for two hours on most days will likely continue to be two hours out of range in future days. However, there are other sets of patients who fluctuate more widely on a day to day basis which are more difficult to predict. With the set of 31 features and 8,000 patients after cleanup and analysis, multiple models were tested while still focusing on explainability.

Firstly, by building an XGBoost Decision Tree Linear Regression model which predicted the exact change in time out of range for the test sets for each patient. Then the depiction of feature importance through a bar graph sorted by most important feature was created. This graph helped us conclude how best to use this information and what to test in future iterations. Following this, hyperparameter tuning on this regression model to see whether this would affect model performance. Afterwards, a similar XGBoost Decision Tree Classification model was built by changing the numerical target variables to categorical variables for an increase, stagnation, or decrease in time out of range of glucose values. This was to test the theory that decision trees work better with categorical target variables and see if this increased prediction accuracies. The feature importance charts were generated for all models built in order for users and researchers to understand which features had the most impact on glucose values. Hyperparameter tuning was once again performed on the classification models with little impact on the final metrics.

The information discovered through these preliminary studies and models led to us building other models to test out theories and gain more evidence to support our research. With the first rounds of modeling, observations made about important features such as a patients' time out of range on a given day and sex have large impacts on their glucose values.

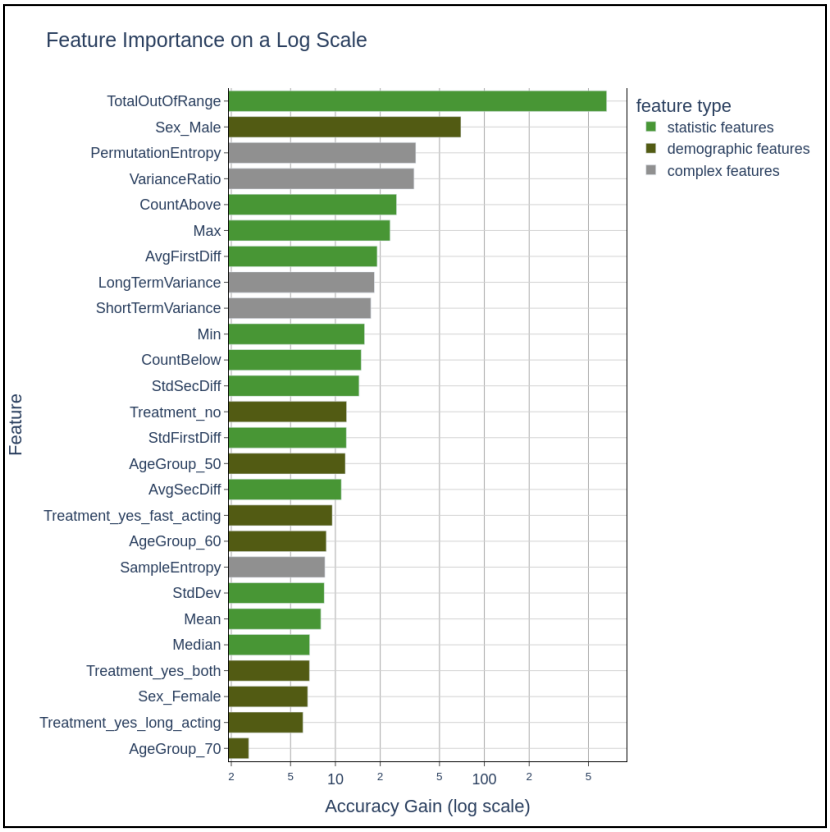

Fig. 3: Made with Plotly, a ranking of how important the features are to the XGBoost Classification model with hyperparameter tuning. Feature importance is measured in Accuracy Gain and displayed on a log scale, differentiating the three types of features through color variation.

Because of this, models were created to remove a patients' time out of range features. This did not decrease performance by much because the same information would be encapsulated with the "time above range," "time below range," mean, maximum, and minimum features. Afterwards, models removing the total out of range, mean, maximum, and minimum features were created to test the same type of data's impact on model performance.

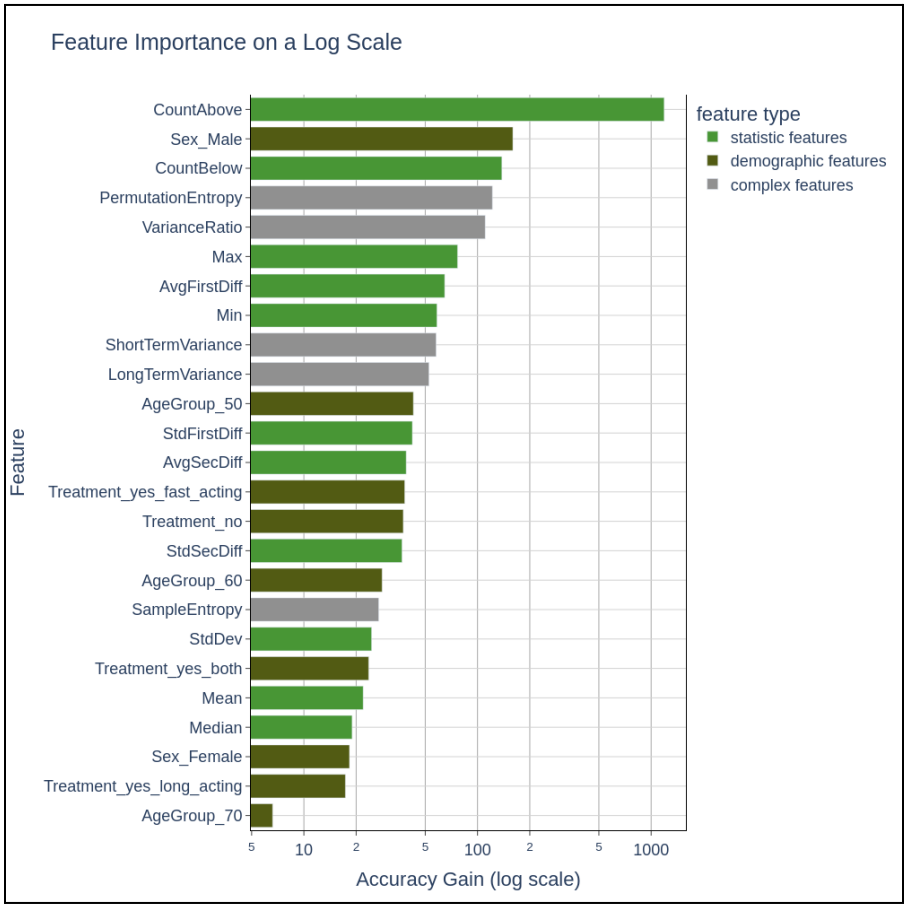

Fig. 4: Made with Plotly, a log-scale ranking of how important the features are to an XGBoost Classification model in which the "time out of range" statistic feature was removed.

The performance decreased, however, not significantly. This is likely due to the fact that this value based information would be encapsulated in other features still present in the model. Similar modeling was performed with the sex feature removed and found similarly that it did not affect performance significantly and the same information was likely captured with other features. To be more detailed: cross fold hyperparameter tuning was used to optimize model performance, and accuracy did indeed increase. A total of 1,215 parameter combinations (3,645 models) were built and tested using Pyspark's Parameter Grid Builder, creating the best performing model with the highest accuracy and F1 scores of the lot.

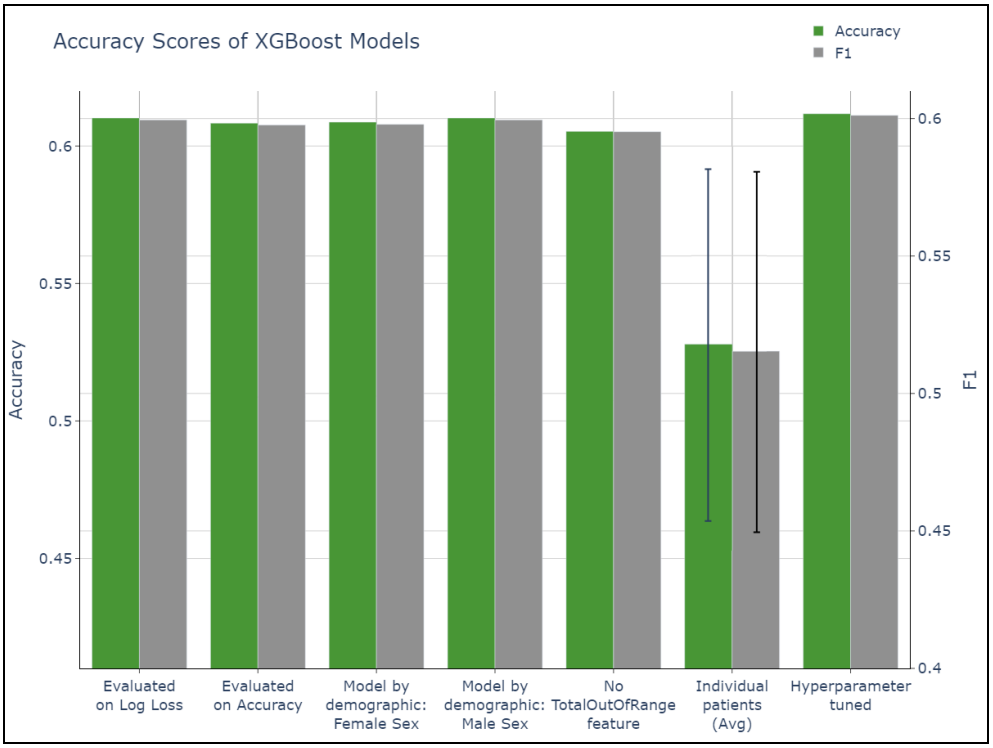

Fig. 5: Bar chart made in Plotly to compare the scores from six XGBoost classification models for predicting patients' change in time out of healthy glucose levels. The *y*-axis on both sides begin above 0.

Furthermore, separate models were created using only statistical features and found that the additional structural features helped glucose trend predictions significantly. The information that the structural features provided added a dimension of trend over time which was not being captured as efficiently with only statistical features, which are generated on a day to day basis. Lastly, individualized models were created for 25 patients using only their own data and saw scores similar to the generalized models.

Model Development

In order to compare whether models at different levels of granularity would have feature or accuracy improvement, models were created at the overall one fits all method, removing some features, models at the biological sex level and at the most granular level being at the patient level.

With regards to the one fits all models where all of the training data was used for the classification and regression models, the best models were the XGBoost Classification and XGBoost Regression models that had hyperparameter tuning using cross fold validation. In comparison, in order to see if the variable total out of range was suppressing other features from being considered important in terms of information gain, the total out of range variable was removed, however there was no performance improvement or feature shuffling in terms of importance. The same was done for removing total out of range, count above and count below features however, again there was no performance improvement feature shuffling that occurred. With the categorical feature of biological sex male, the separation of biological sex was taken

into account so models that were trained on only male and only female data were created, however there was also no performance improvement on the models. As a result, in order to see if these models needed to be trained on patient level data, models were trained on each individual patient using only their data with the trade-off being the models were very customized to that patient however less training data was used due to granularity of the issue. There were also no performance metrics that improved using this technique. The best models were the one fits all models where all patients training data was used to create the classification and regression models, which shows the importance of having large amount of training data.

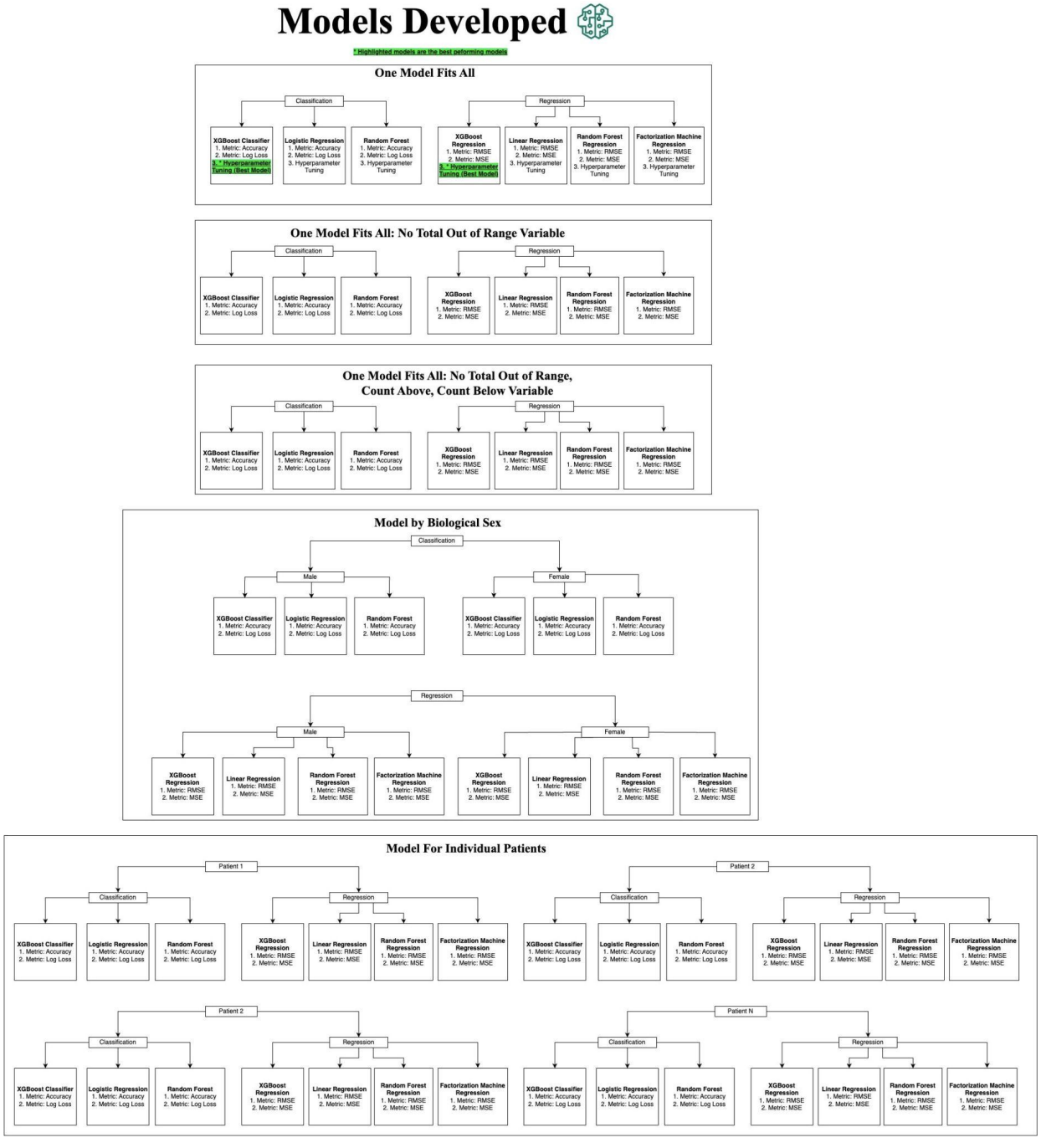

Fig 6: Tree figure of the different types of models tested in order from general to more granular models in terms of training data ranging from all patients to the patient level.

Solution Architecture, Performance, and Evaluation

The architecture for these models was created to be used for data at scale using the PySpark machine learning library and wrapper for XGBoost which made large data processing possible. Model performance metrics were evaluated in a few different ways for this project. With regression models, being evaluated on RMSE and R^2 values. Conversely with classification models, accuracy, precision, recall, and F1 scores were evaluated to measure overall model performance.

Regression Models				
	FM	**LinReg**	**RFReg**	**XGBReg**
RMSE	27.8188	27.7562	27.8469	26.8886
R2	0.3598	0.3627	0.3585	0.4019

Classification Models			
	LogReg	**RFClas**	**XGBClas**
Accuracy	0.5662	0.5675	0.6102
F1	0.5590	0.5233	0.5995

Fig. 7: Table of the scores from the four regression and three classification models for predicting patient's change in time out of healthy glucose levels.

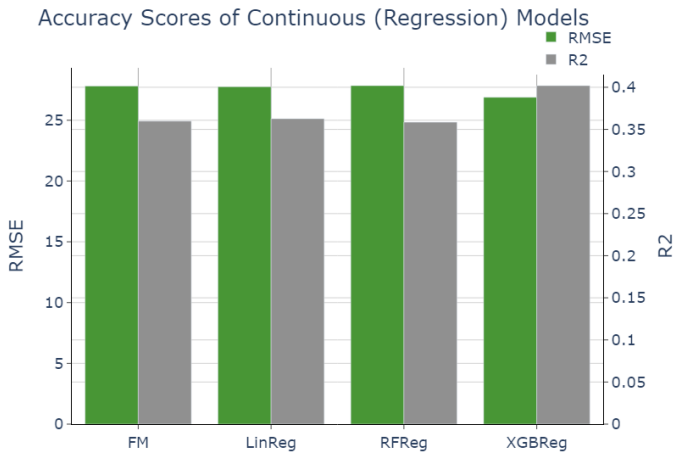

Fig. 8: Bar chart made in Plotly of the scores from the four regression models for predicting patients' change in time out of healthy glucose levels.

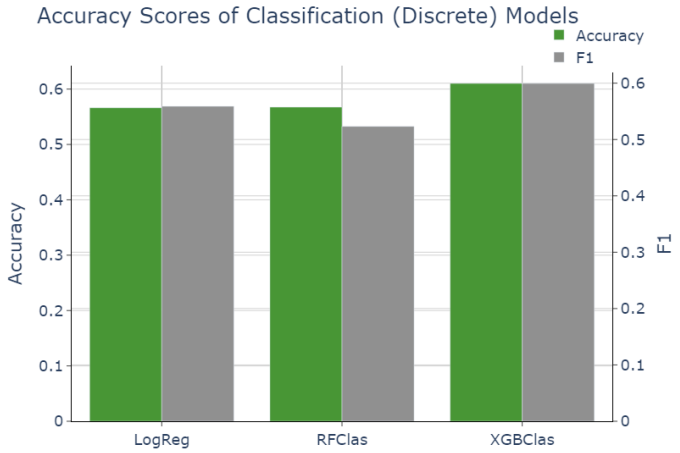

Fig. 9: Bar chart made in Plotly of the scores from the three classification models for predicting patients' change in time out of healthy glucose levels.

Feature importance was measured as the average information gain per split in the decision tree model. See Fig. 2 for an example of this.

Due to the data being housed on the UCSD Nautilus cloud provider with strict restrictions, there was no movement of the data to AWS to do any scaling or processing and therefore did not need budget management. The project's results for top performing models, which provided the bulk of these findings, are shown above with their respective performance measures.

Web Application

With these findings in hand, an application was developed that can be used by patients to take back agency of their diabetes management and improve their understanding of which metrics cause them to be in range or out of range. It was envisioned that the application would be used when the users wake up. Users can log into their app and receive insights on how their glucose levels have changed compared to their previous day. When the user arrives at the application, they are prompted with a login screen and information on DiabeatIt. Once logged in, the application selects the users data, trains the high accuracy XGBoost model that was developed with a full year of the patients data for this project, and outputs a prediction of either improving time within range, decreasing time within range, or staying consistent compared to the previous day's value. While a prediction is very helpful, what prompts actions and furthers understanding is providing insights on what features had the highest impact in the chosen prediction. To present this knowledge, a library called Eli5 was utilized. Eli5 has functions that allow for the ingestion of a model and testing data, and provides single prediction feature importance. By using the trained model and the user's previous day's data, Eli5 extracted the feature importance and was able to display that to the user. To make this accessible to all users, an explanation was provided per feature about what this feature importance could tell you about your body and actions to consider to improve or maintain your trends. In addition to the prediction and insights, the ability to maintain a level of transparency with users was very important. By stepping into the prediction space in the medical field, transparency is very important and needs to be taken seriously. With this in mind, the prediction confidence for that

single prediction is provided to the users. This allows users to make a personal judgment on how, or even if, they will take action based on the prediction. By creating this app, users are granted the ability to prepare for their day of glucose fluctuations with expectations, as well as track which features often affect different types of their personal predictions. This trend tracking can lead to actions to improve their time within healthy range.

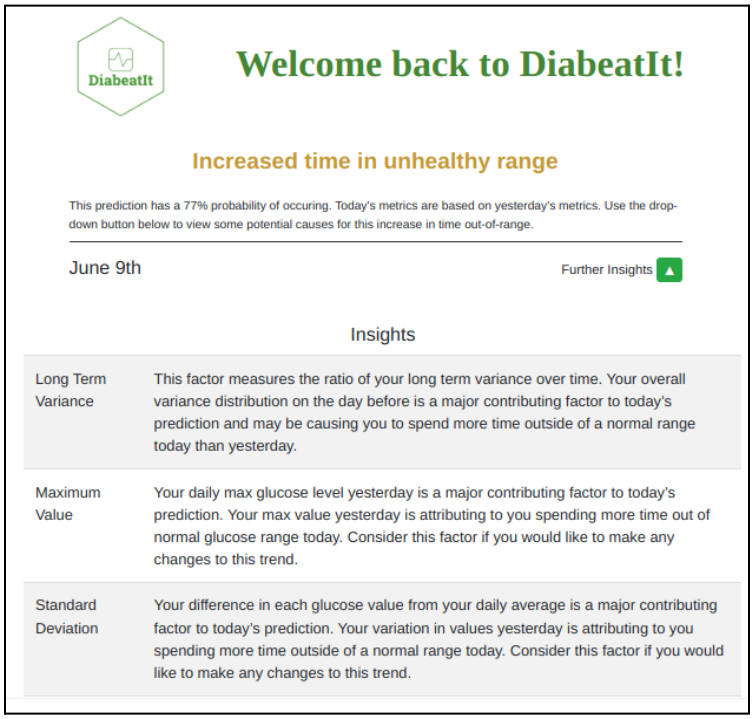

Fig. 10: A visualization of the application user-interface.

Conclusions

Predicting glucose levels in diabetes patients is a field of study that has gained popularity with the advent of machine learning models and predictive systems. Much of the other research in this field, however, focuses on glycemic event prediction with high accuracy which leads to a lack of explainability. Using explainable models and calculating distinguishable features is what sets this project apart from the others. Through the use of XGBoost Decision Tree models and time series data analysis, this project found that patients' past history of time spent out of normal glucose range, their sex, permutation entropies, and ratio of their glucose levels; variance are major factors in determining glucose fluctuations from day to day. It is still important to keep in mind that this is a complex biological system and therefore many other features not captured by this model contribute to fluctuations and data trends, giving future data scientists the opportunity to add other features and increase performance. The accuracy score of this project's final multi-classification model is 61.12% with feature explainability and three targets. This showcases a higher level of order in the model than random guesses, and breaks significant new ground in exploring long-term horizon prediction. Results found here may be improved with yet more experimentation and analysis.

References

"Distributed XGBoost with Pyspark." *Distributed XGBoost with PySpark - Xgboost 1.7.5 Documentation*, xgboost.readthedocs.io/en/stable/tutorials/spark_estimator.html. Accessed 20 Mar. 2023.

"Diabetes." *World Health Organization*, www.who.int/news-room/fact-sheets/detail/diabetes. Accessed 8 May 2023.

Goyal R, Jialal I. Type 2 Diabetes. [Updated 2023 May 8]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2023 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK513253/

Gecili, E., Huang, R., Khoury, J., King, E., Altaye, M., Bowers, K., & Szczesniak, R. (2021). Functional data analysis and prediction tools for continuous glucose-monitoring studies. *Journal of Clinical and Translational Science, 5*(1), E51. doi:10.1017/cts.2020.545

Kohnert K-D, Heinke P, Vogt L, Augstein P and Salzsieder E (2018) Applications of Variability Analysis Techniques for Continuous Glucose Monitoring Derived Time Series in Diabetic Patients. *Front. Physiol.* 9:1257. doi: 10.3389/fphys.2018.01257

Laverty B, Puthezhath Jayanandan S, Smyth S. Understanding the relationship between sleep and quality of life in type 2 diabetes: A systematic review of the literature. *Journal of Health Psychology.* 2023;0(0). doi:10.1177/13591053221140805

Millson V, Hammond P (2020) How to analyse CGM data: A structured and practical approach. *Journal of Diabetes Nursing* **24**: JDN135

"PySpark Overview¶." *PySpark Overview - PySpark 3.4.0 Documentation*, 18 Apr. 2023, spark.apache.org/docs/latest/api/python/index.html.

Mary Martin, Elizabeth Chun, David Buchanan, Rucha Bhat, Shaun Cass, Eric Wang, Sangaman Senthil & Irina Gaynanova. (2021, April 27). irinagain/Awesome-CGM: List of public CGM datasets (Version v1.1.0). Zenodo. DOI 10.5281/zenodo.3895210

Link to the Library Archive for Reproducibility: https://doi.org/10.6075/J0XW4K0B