Diabetes in USA: Public Health Led by Data

Temesgen Fekadu, Jack Lynn, Hayden Muscha, & Sam Wainright

Dev10 Data Science & Engineering Capstone

August 18, 2022

I. INTRODUCTION

Non-communicable diseases dominate disease burden in the developed world, of which diabetes is one of the leading causes of disability-adjusted life years (DALYs). Diabetes is especially prevalent in United States, contributing the seventh largest DALYs of any public health burden. Unlike other leading health burdens heart disease and cancer, diabetic individuals are more burdened by years of healthy life lost due to disability (YLDs), making up 5.50% of American YLDs as compared to making up 2.64% of deaths. [1] Such a large disease burden necessitates refined public health programs that address diabetes awareness and long-term care.

To advocate for diabetes as a public health burden, one must determine the communities most affected by it, the underlying medical risk factors that increase the likelihood of its development, and its current maintenance and treatment methods. This capstone project will use a three-pronged data science and engineering approach to create a holistic narrative around the disease burden of diabetes in United States from a public health perspective. First, CDC [2-3], U.S. Census [4-5], and U.S. Department of Agriculture [6] data illustrate through visualizations who is most affected by diabetes, including special interest explorations of region, income, food security, and level of education. Second, machine learning techniques use Rui-Ci Health Center's diabetes and medical characteristics dataset [7] to predict the underlying physical characteristics that coincide with diabetes diagnoses, emphasizing the connection between community demographics and health manifestations. Third, blood glucose traces from JAEB Center [8] are streamed to represent Type-1 diabetic patients' blood glucose traces recorded in real time; deep learning synthesizes these streamed data to forecast blood glucose levels 30 minutes into the future, which can be used for warning patients of hazardous blood glucose spikes. This capstone's approach exploits the use of cheap data science techniques to characterize diabetes care pre- and post-diagnosis. The findings of this project should be utilized by public health agencies to target diabetes public awareness programs and refine how diabetes in maintained in United States.

This capstone project will answer the following questions:

1. Which demographics are most likely to develop diabetes in the US?
2. What measurable bodily attributes suggest the presence of diabetes?
3. How are diabetes patients' blood glucose levels tracked in real time?
4. Are certain regions of the United States more affected by diabetes than others?
5. Does food scarcity impact diabetes incidence?
6. Can we predict diabetes diagnoses based on readily available medical vitals, such as blood pressure, mineral levels, and body mass index?

7. Can we predict blood glucose levels of Type-1 diabetes patients at least 30 minutes ahead to warn about incoming hazardous spikes?

II. DATA SOURCES & ETL

Seven datasets pertaining to diabetes and United States demographics are used to create a narrative of diabetes burden. Visualizations characterizing the demographics of diabetes in United States make up five of the datasets used. From the Centers for Disease Control and Prevention, two datasets–(1) National Health and Nutrition Examination Survey (NHAINES) [2] and (2) diabetes prevalence among state, race, and ethnicity [3]–describe general health and demographic characteristics of those with and without diabetes in the United States. From the U.S. Census Bureau, two datasets–(1) educational attainment by state [4] and (2) income brackets by state [5]–contribute state-grain education and income demographics to be correlated with diabetes data. From the U.S. Department of Agriculture, their food security by state data [6] is used to correlate diabetes to food access. Additionally, two much larger datasets are used for the data streaming, machine learning, and deep learning portions of this project. First, the Rui-Ci Healthcare medical vitals and diabetes diagnosis dataset [7] is used as the training and testing data for machine learning models tested later in this project. Second, the JAEB Center for Health's Continuous Glucose Monitoring (CGM) readings [8] provide data for streaming real-time blood glucose levels of Type-1 diabetes patients and serve as the basis from training the deep learning models of this project.

We extracted all data from their respective websites as CSV to eventually be transformed into an SQL database stored on Azure SQL Databases. A full description of the extract-transform-load (ETL) process (*Fig. 1*) can be found in the GitHub corresponding to this project. We conducted our transformation of each dataset using Python libraries Pandas and Numpy in Python notebooks. The most common transformations are removing null fields, dropping unnecessary columns, managing inconsistent or unusable typing, handling outliers, and schema simplification. After completing initial transformations, each dataset is exported to CSV files and uploaded to our group's Azure Data Lake. Using PySpark and Kafka in Azure Databricks, the datasets go through a second round of transformations to make their structure consistent with SQL schemas before

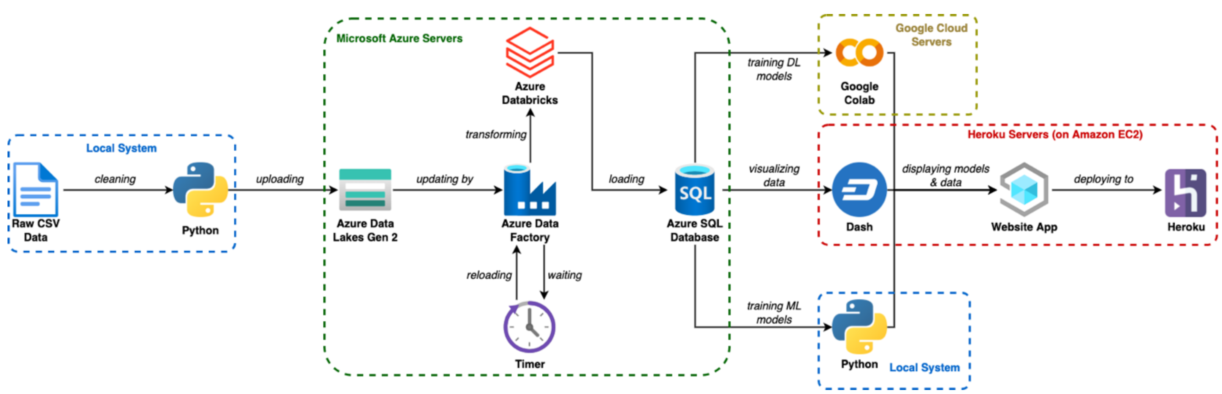

Figure 1: Network flow diagram description of ETL and analysis process of data in this capstone project. The ETL process is entirely contained within local and Microsoft Azure systems, whereas all visualizations and learning models are contained in local systems and Google Cloud Servers. Note that the original breadth of this project intended to deploy the website-like dashboard to Heroku, but time constraints limited this goal.

being loaded into our group's Azure SQL Database. Additionally, a data stream in Azure Data Factory runs the loading process repeatedly every 24 hours to ensure the SQL data is up to date.

III. DATA VISUALIZATIONS

III.A. Dash Dashboard

For the dashboard, Plotly's Dash is utilized with bootstrapping components and HTML to create a set of website-like pages. Markdown text, images, menus, and graphs then populate the pages using Dash's Python library. The visualizations in the dashboard are created with the aid of both Plotly Express and Plotly Graph Objects. Some visualizations are made outside of the Dash environment and imported in as PNG files to be placed inside of the HTML layout of the Dash pages. For the graphs dynamically created on pages, we chose to use PyMySQL to query the remote Azure database into Pandas DataFrames, which we then transformed into Plotly visualizations.

For the homepage, we decided to include an overview of diabetes as well as the goals of this project to help inform the viewer about information we will present. We also included a choropleth graph, in which the color indicates diabetes prevalence across different states in United States. The map also includes a sliding bar for changing year of reference. We also included a line graph that shows diabetes prevalence of the nation and individual states over time. This visually is included to show more concretely the changes in prevalence over time.

For the "Predicting Diabetes" page, we included an explanation of the machine learning model to predict diabetes diagnosis based on Rui-Ci patient data. [7] Included are ANOVA scores of all features evaluated, showing which factors have the greatest influence on the model. We also include a confusion matrix to assess the sensitivity and specificity of the model. We also include a ROC curve of the model to show the trade-off between sensitivity (or TPR) and specificity (1 – FPR) of our model.

For the "Indicators and Factors" page, we included four histograms showing the distribution of people who are at risk for diabetes, prediabetes, and diabetic risk as compared to demographics for education, body mass index (BMI), frequency of eating out, and income bracket. These histograms are used to both reinforce the findings regarding BMI of our machine learning model as well as illustrate the demographic commonalities of those at risk of developing or currently having diabetes. Education is utilized to view if an investment into overall education in the U.S. could potentially have impact on diabetes outcomes. Income is utilized as an indicator of potential target neighborhoods where more resources would be helpful in diabetes prevention. The frequency of eating out is used to evaluate if individuals that have developed or may develop diabetes are struggling with access to a reliable quantity of affordable, nutritious food.

For the "National Examination" page, we included three choropleth maps that show income, education, and food insecurity across the nation on a state-by-state level to highlight the regions that are performing poorly regarding the metrics addressed on previous page. Additionally, the bottom of the page contains a call to action that can be taken to help combat the increase in diabetes prevalence.

For the "Personal Glucose" page, we included a live updating line graph that shows blood glucose monitoring simulating in real time. This is included to demonstrate the live data streaming aspect of the project and to present the possibility of helping those with blood glucometers manage their blood sugar levels and their diabetes.

For the "DL Prediction" page, the overall structure is a proof-of-concept of how the deep learning model predicts blood glucose traces. Included is a brief description of the model itself and four parameter graphics illustrating how the model is tuned. Included with each parameter are three graphs that describe its performance. The first graph is a line or bar graph describing loss in comparison to the training metric. The second graph is a line or bar graph describing training time in comparison to the training metric. The third metric compares the trained models against the truth values. Navigating between these training metrics is completed using a dropdown menu.

III.B. Data Streaming

To address real time blood glucose monitoring, we created a blood glucose tracking interface by utilizing Kafka's producer/consumer tools in Azure Databricks (*Fig. 2*). We created a PySpark DataFrame using one patient's data from the cleaned CGM readings CSV file and sorted the readings in chronological order. We then created a producer that goes through each row of the data frame and sends one message every 5 seconds containing that respective row. While the original CGM device took measurements every 5 minutes, we used 5 second intervals for easier development and demonstrations; however, this metric can be easily altered. After creating the producer, we created a consumer that would take in the messages from our producer and load them into our Azure Database. Lastly, we created a pipeline that would run the producer and consumer upon triggering. We added a delay on the consumer to allow the producer enough time to unmount and mount onto the data lake before the consumer started reading messages to avoid timing out.

Now that our Azure Database was receiving new glucose readings every five seconds, we created a glucose tracker interface in the dashboard. We started by using PymsSQL to query the database to select the top 24-hours-worth of readings ordered by the timestamp in descending order. Using this query, we created a Plotly line chart with animation on and used an interval of 5 seconds to update the chart in a callback function. The live glucose tracker functions fully, refreshing the data from the Azure database every 5 seconds with new CGM reading entries from the producer/consumer. We added a couple more callback functions to the dash script to update the lowest and highest glucose readings of the last 24 hours.

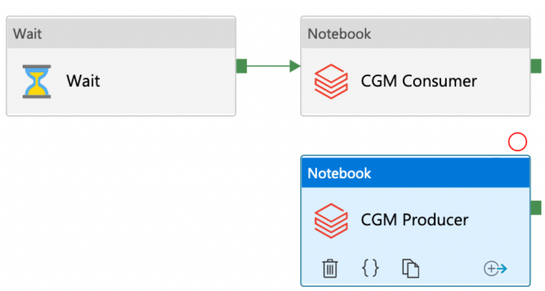

Figure 2: CGM data stream pipeline. When called, the producer starts immediately while the consumer must wait a time delay first.

IV. MACHINE & DEEP LEARNING

IV.A. Machine Learning

Machine Learning (ML) is a type of artificial intelligence that extracts patterns out of raw data and focuses on learning from experience without being explicitly programmed. The need for machine learning in medical spaces shows benefit in domains with little or dynamic data, and the application of it can be used to solve problems in error detection and prevention. The goal of our ML model is to determine if, given the vitals and demographic information of the patient, a Type-2 diabetes diagnosis can be predicted.

ML models utilize the Scikit-Learn library and Python scripting within Visual Studio Code. Scikit-Learn is built upon SciPy, NumPy, and MatPlotLib, making it ideal for ML algorithms like classification, regression, and clustering. The library also provides tools for evaluating models, tuning hyperparameters, and preprocessing data. Since Scikit-Learn is open source and commercially available, ML tasks are accessible to the public.

The general template of deploying a ML algorithm for this capstone begins with finding an appropriate dataset. Following this, a large part of the process involves understanding the data so that an appropriate model is used. The process begins with exploratory data analysis and descriptive statistics prior to moving into ETL. Following this, algorithms are evaluated, tuned, and finalized by making predictions on a validation dataset.

Exploratory data analysis determined that the algorithm used should be utilize supervised learning focused on classification-based tasks. Models are evaluated by their ROC Curve and G-mean. Finally, the dataset used presented an imbalance in target classes. An appropriate model for our dataset is one that is capable of handling severely skewed class distributions and able to distinguish between the two classes

IV.B. Deep Learning

Deep learning (DL) is especially adept at learning the complex topographies of time-series data, so we chose to use DL to forecast blood glucose levels and its especially noisy data. The goal of the DL model is to warn diabetes patients if they are likely to have hazardous (180 mg/dL or greater) and dangerous (300 mg/dL or greater) blood glucose spikes [9] within the next 30 minutes. Blood glucose forecasting is framed as a supervised, regression, time-series DL problem, so we decided that recurrent neural network (RNN) models are a strong choice. We also decided to make the models using the TensorFlow library in Google Colab. TensorFlow is an extensive DL library integrated into Python that can run on GPU processors, making it up to 85% faster than if it ran on CPUs. [10] This acceleration is especially necessary because DL models can have complex architectures of neural network layers that require high computation and memory costs. Google Colab is a natural partner to Tensorflow, as it has free-to-use GPU-accessible cloud computing and a built-in environment to handle Tensorflow out-of-the-box. The dataset used for training and testing this model is the CGM blood glucose traces [8] in conjunction with the real-time CGM data stream.

The core structure of all DL models trained for this capstone follow the same input and output schema. The input is a window of a specified number of current and previous blood glucose level datapoints. The output is a single forecasted blood glucose value a specified amount of time into the future. Training and testing are done by taking previously recorded blood glucose traces and splicing them into input and output groups as previously described. DL models are trained using TensorFlow's standard batch training for a specified number of epochs, in which backpropagation is completed after each epoch.

We completed four individual parameter tests to refine the DL model to create our optimal model; each test is evaluated using mean squared error as loss. Due to DL models' long training times and time constraints, individual parameters are refined one at a time as opposed to integrated searching mechanisms. The first test varies the input window size between 5 and 50, increasing by 5 each time. The second test varies the forecast time between input and output between 5 and 120 min, increasing by 5 min each time. The third test varies training time from 1 to 50 epochs, particularly focusing on when overfitting occurs. The final test varies the layer composition for the number of long short-term memory (LSTM) and gated recurrent unit (GRU) layers; as these are the current industry standard for RNN layers, we only tested these types of neural network layers, though dense, dropout, and flatten layers are used in all models. The optimal model is composed of what is deemed to be the best outcome of each optimization test, which is evaluated based on minimization of loss, training time, and faithfulness to the goal of the algorithm.

The optimal model is then compiled into a basic notification system that takes in the CGM real-time data stream to evaluate if the patient should be notified for a forecasted hazardous or dangerous blood glucose spike. The model will take in the last 20 data points from the data stream and pass it through the optimal DL model. If the prediction is greater than 300 mg/dL, the model will print an alert stating that the patient is at extreme risk; if the prediction is greater than 180 mg/dL, the model prints a warning stating that the patient's blood glucose is dangerously high. This basic framework for notification can then be adapted to the specific messaging system of a CGM monitor, thus making it smarter.

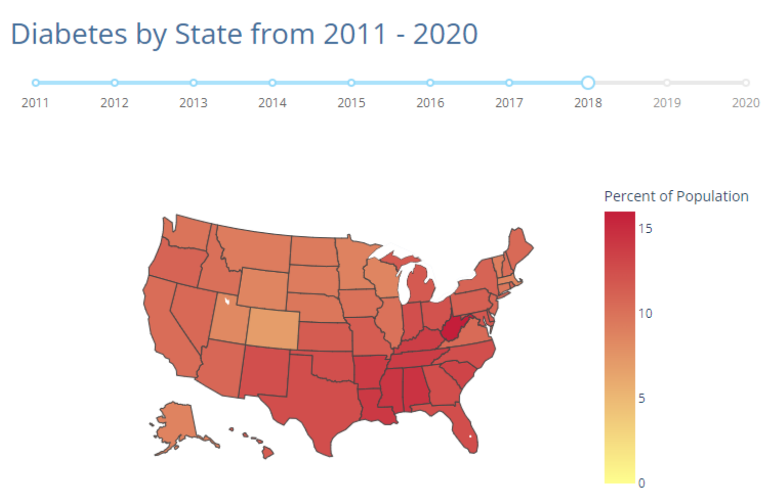

Figure 3: Diabetes prevalence by U.S. state.

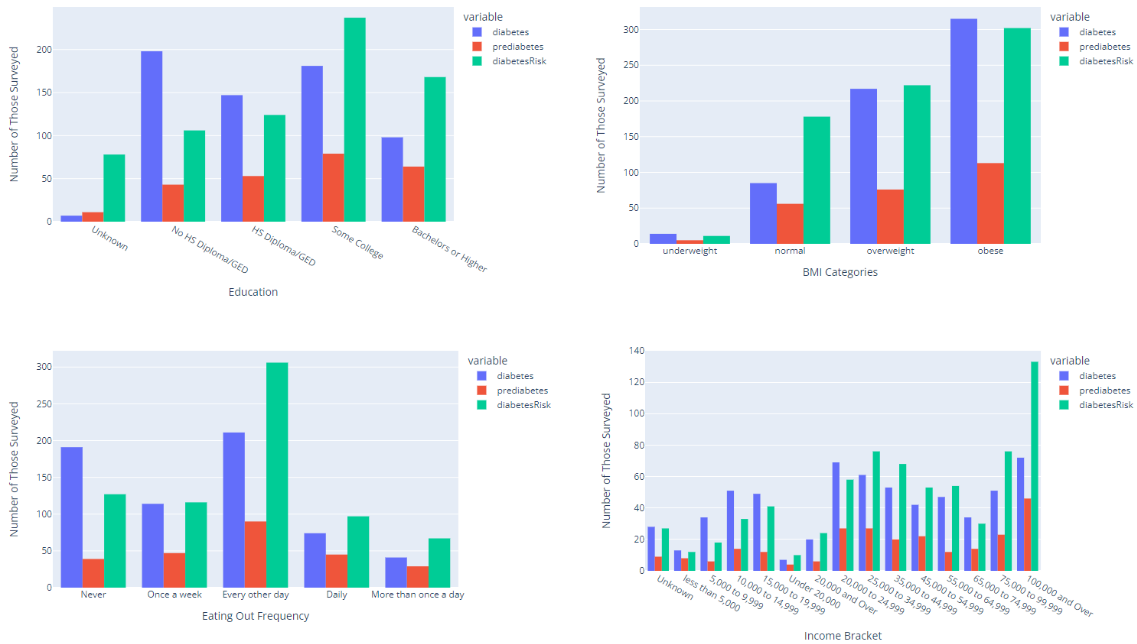

Figure 4: Diabetes incidence of different demographic groups, including education, BMI, frequency eating out, and income bracket.

V. RESULTS

V.A. Demographic Exploration

Diabetes prevalence across the United States is not a uniform distribution (*Fig. 3*). The Southeast has consistently the highest diabetes prevalence of all states, with West Virginia being the highest. The West Coast, parts of the Midwest, and parts of Northeast are noticeably higher than the rest of states. Colorado has an outstandingly low diabetes prevalence, noticeably dissimilar to its neighboring states.

Certain demographic groups have far higher incidence of diabetes (*Fig. 4*); in particular, BMI is one of the strongest indicators of diabetes. Particularly overweight and obese people are at higher risk of being diabetic, prediabetic, and at diabetic risk. Additionally, education and income are both factors that correlate to development of diabetes. Those who do not have a bachelor's degree or higher and those who make less than the U.S. median income are both higher risk of diabetes, potentially indicating that those with lower access to healthcare have higher diabetes risk. Finally, those who eat out more than 3 times a week have much higher incidence of prediabetes, indicating that diet potentially influences diabetes incidence. In whole, addressing access to whole, nutritious food and healthcare are strong needs of diabetes control in United States.

V.B. Machine Learning

The first optimization for training the ML model is feature selection using ANOVA F-tests, which measures the variations between the two categories (*Fig. 5L*). The results of the test show which features are independent of the target variable of diabetes and can be removed from the dataset to reduce noise and improve the processing time. Features with a higher score are those that are

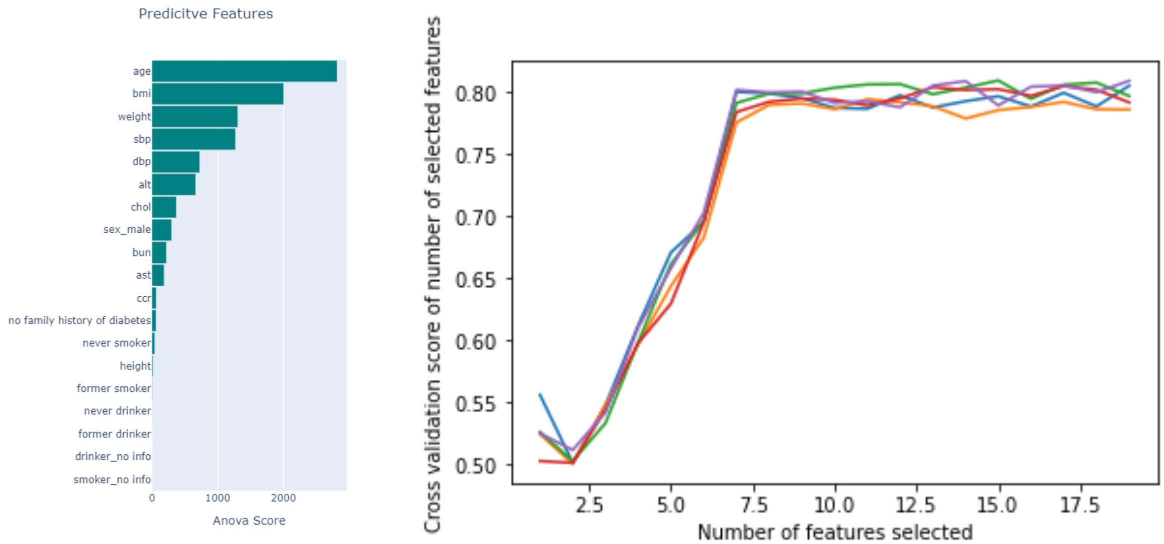

Figure 5: *(Left)* ANOVA F-tests for feature selection ML model. (*Right*) Optimization number of features in ML model.

farthest from the mean. The features that score closer to 0 represent and equal variance and therefore has very little impact on the target variable. Results of the F-test show that the most impactful variable on a diabetes diagnosis is age, followed by BMI, blood pressure, and cholesterol. The ideal number of features is selected through recursive feature elimination wrapped with a random forest classifier and determined that optimal number would be 7 for the selected Linear SVC model (*Fig. 5R*).

Evaluation of the model chosen focused on the performance, measured by the ROC AUC (*Fig. 6L*) and recall. Following tuning of the Linear SVC model with the selected features, the model displayed an ROC AUC score of 0.84, suggesting that the model is capable of distinguishing between the two target variables: (1) patients with diabetes, and (2) patients without diabetes. The confusion matrix (*Fig. 6R*) visualizes the number of predictions the model made towards each class and through this, recall on the positive class came close to 80%. One major issue with the training dataset, however, is that the proportion of people with the disease is around 2%, meaning

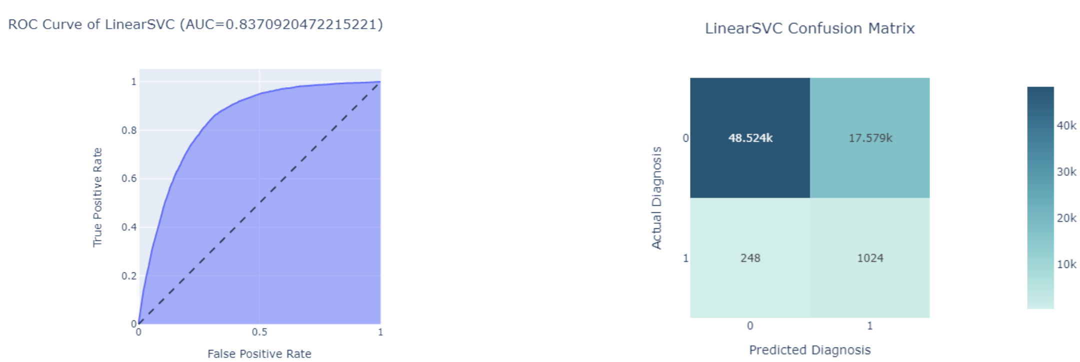

Figure 6: *(Left)* ROC AUC curve of optimal ML model. (*Right*) Confusion matrix of optimal ML model.

Figure 7: Snapshot of CGM blood glucose real time streaming, including daily high and lower blood glucose alerts.

resampling and sensitivity boosting had to be used to prevent a ML model that never predicts diabetes diagnoses. As a result, while sensitivity remains high at 80.5%, specificity drops to 73.4%, meaning many people without diabetes are getting falsely classified. Therefore, the tool is best used as a vetting device for suggesting if people should get tested for diabetes but should never be used in diagnosis without a second opinion.

V.C. Data Streaming

The main visualization used for our CGM tracking interface is a dynamically updating line graph, as seen in *Fig. 7*. On the x-axis is the time of day; on the y-axis is the blood glucose level. We also included minimum and maximum glucose levels in a 24-hour window below the line chart. Being able to see real time blood glucose levels, people living with diabetes can now notice different patterns and certain times of the day when glucose levels are high and low. They can see the range of the glucose levels they have, the impact of diet, and the effectiveness of insulin. The data is important for a diabetic patient to be able to measure how severe their case is and how well their treatment is working.

V.D. Deep Learning

Several DL models with different parameters are trained to forecast blood glucose levels using CGM data; the optimal parameters of these tests are then compiled into an optimal DL model. Mean squared error and training time are used to evaluate the efficacy of each parameter. Input window sizes (*Fig. 8A*) are varied to determine how many prior readings are needed to make an accurate prediction. Any window size below 10 previous points results in more than 12-fold increase in loss; after 10, loss still decreases with window size, though not as sharply as before. We chose a window size of 20 for the optimal model because it balances training time, loss reduction, and minimization of overfitting. Forecasting times, which describes how long into the future the blood glucose prediction is forecasted, (*Fig. 8B*) show a nearly linear increase in loss with amount of time forecasted into the future. We chose a 30-minute forecast because it gives

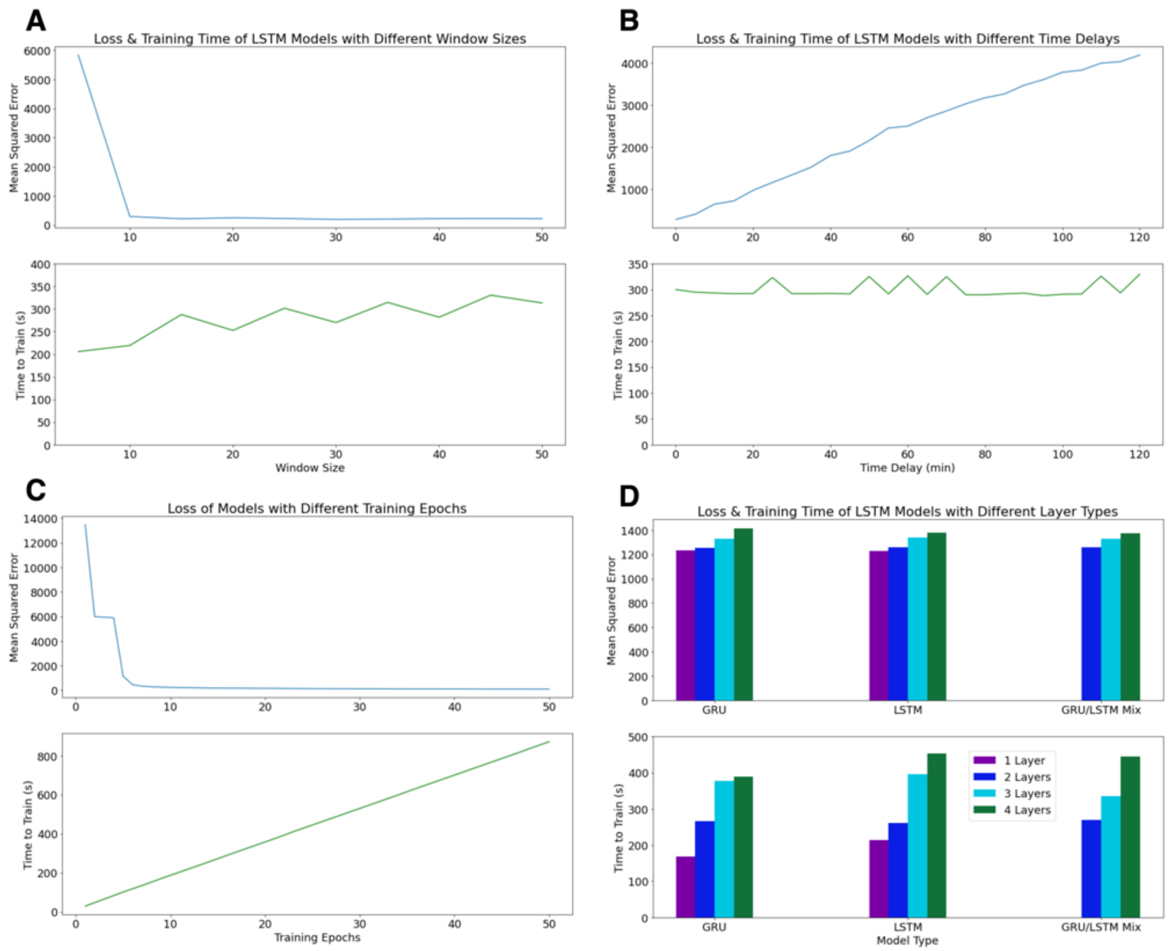

Figure 8: Loss and training time compared to training parameter: (A) window size, (B) time delay (forecasting), (C) training epochs, and (D) layer types and number.

sufficient forecasting time to make the forecasts useful to the user without dramatically increasing loss. Number of training epochs (*Fig. 8C*) is evaluated to determine the number of epochs necessary for sufficient loss reduction without overfitting. Number of training epochs increases the time necessary of train linearly; the number of epochs until models are sufficiently trained appears around 25 to 30 epochs. First, there is a large drop off around 5 epochs because the parameters are randomly assigned initially. We determined 30 epochs to be the ideal training time because there is still sufficient loss reduction, though less dramatic, between 10 and 30 epochs. Finally, model composition (*Fig. 8D*) is evaluated for what best handles the blood glucose traces. GRU and LSTM layers do not perform particularly better than the other, though GRU layers tend to be slightly faster. The number of layers increases loss, though there is not a strong distinction between one and two layers; however, more layers may be undertrained because only 10 training epochs are used to train these tests. Ultimately, we chose to use 1 layer of LSTM and 1 layer of GRU in the optimal model to harness the advantages of both RNN layer types.

The optimal DL model (*Fig. 9A*) is composed of 1 LSTM layer, 1 GRU layer, 3 dense layers, 2 dropout layers, 1 flatten layer, 2 ReLU activation functions, and 1 linear activation layer. The LSTM and GRU layers are optimized features whereas all other layers are the recommended layer

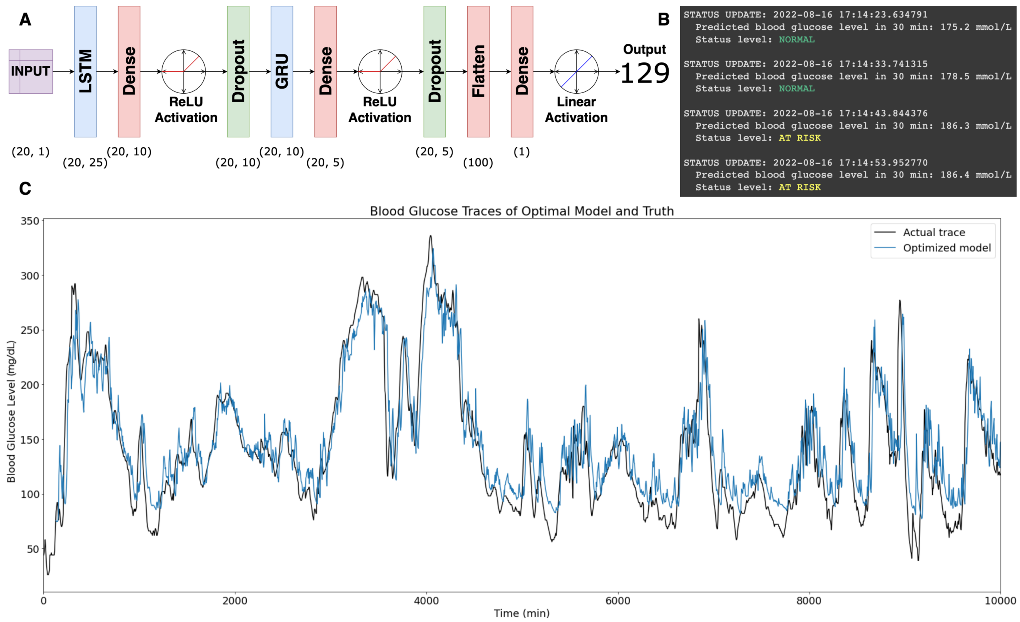

Figure 9: (A) Layer composition of optimal DL model. (B) Sample outputs of test alerting systems. (C) Comparison of optimal blood glucose trace to actual trace.

compositions of supervised time-series regression models. In particular, the dense and flatten layers format the inputs and output layers, the dropout layers allow for better training, and the ReLU and linear activation functions are scaled to work well with regression models. The optimal model has an overall loss of less than 1000 while forecasting 30 minutes into the future. A sample trace of the optimal model is compared against truth in *Fig. 9C*. The model is particularly adept at catching the general shape of the blood glucose traces and is especially strong at times of rapid incline and decline. It has some issues catching noisier areas and the specific values of peaks and valleys, as it often underestimates peaks and overestimates valleys. However, these deficiencies do not invalidate its utility at tracking blood glucose levels in real time. *Fig. 9B* demonstrates a sample program taking in real-time data from the CGM stream and predicting future blood glucose levels using the optimal model. With simple interface implementation, the model can be used to warm patients when their blood glucose levels are projected to get hazardously high, as seen with the text warning highlighted in yellow.

VI. CONCLUSION

This capstone project succeeds in its goal of creating a holistic narrative around diabetes in the United States. Data visualizations reveal that the most affected groups by diabetes are those below the median income, have less than a Bachelor's education, live in the Southeast, have high BMI, and have low consistent access to fresh, homecooked meals. Our machine learning model then provided an easy-to-use, cheap, and readily accessible tool that can be the first test people use to determine if they should get tested for diabetes. On the post-diagnosis end, our blood glucose streaming tool paired with our deep learning blood glucose forecasting model suggests a future in which diabetes care can be more automated to the patient and "smarter". Altogether, diabetes is a major, complex public health burden; using data science to United States' advantage can make diabetes public outreach accessible and tailored to those who need it the most and to the most effect.

Our project has also supported the following suggestions for diabetes outreach:

1. The states with higher incidences of food security also trend toward higher incidences of diabetes in the Unites States. As such, investing in projects that focus on creating greater food security in these communities is essential. Contacting local representatives and trying to enact change on a political level could also be a path towards effective and permanent change in diabetes care in United States.
2. The trends in increased diabetes in those below the median income may also trend toward lower access to healthcare. This deficit can be addressed with Supplemental Nutrition Assistance Programs (SNAP), which elevate the quality and availability of food to those who can take advantage of the program. These programs could include more informational sessions and advertisements of diabetes awareness and care within those who use these programs.
3. Our machine learning model serves as a great initial screening tool to encourage people who are at risk of diabetes to get tested. However, it is not a replacement for proper diabetes diagnostic tests but rather an encouragement to get tested in the first place.
4. On the post-diagnosis end, further development and expansion of CGM technology are essential to the future maintenance and control of blood glucose levels. The infrastructure already exists for tracking real-time blood glucose concentrations, so the next step is to make software that synthesizes data smartly to automate care to the patient. The use of machine and deep learning have the potential to remove some of the anxiety and burden of diabetes patients needing to dutifully track their blood glucose levels constantly.

VII. RESOURCES

[1] Institute for Health Metrics and Evaluation. (2019). GBD compare. Data Visualizations. Retrieved August 8, 2022, from https://vizhub.healthdata.org/gbd-compare/

[2] Centers for Disease Control and Prevention. (2017, January 26). National Health and Nutrition Examination Survey. Retrieved August 5, 2022, from Kaggle website: https://www.kaggle.com/datasets/cdc/national-health-and-nutrition-examination-survey

[3] Centers for Disease Control and Prevention. (2022, March 24). U.S. Chronic Disease Indicators: Diabetes | Chronic Disease and Health Promotion Data & Indicators. Retrieved August 5, 2022, from Socrata website: https://chronicdata.cdc.gov/Chronic-Disease-Indicators/U-S-Chronic-Disease-Indicators-Diabetes/f8ti-h92k/data

[4] United States Census Bureau. (2022a, March 17). American Community Survey: B06009: PLACE OF BIRTH BY EDUCATIONAL ATTAINMENT IN THE UNITED STATES. Retrieved August 5, 2022, from Census website: https://data.census.gov/cedsci/table?q=education%20by%20state&tid=ACSDT5Y2020.B06009

[5] United States Census Bureau. (2022b, March 17). American Community Survey: S1901: INCOME IN THE PAST 12 MONTHS (IN 2020 INFLATION-ADJUSTED DOLLARS) U.S. Census Bureau (2020). Retrieved from Census website: https://data.census.gov/cedsci/table?q=S1901&tid=ACSST5Y2020.S1901

[6] U.S. Department of Agriculture. (2021, September 8). Food Security in the U.S. USDA (2019). Retrieved from www.ers.usda.gov website: https://www.ers.usda.gov/topics/food-nutrition-assistance/food-security-in-the-u-s/interactive-charts-and-highlights/

[7] Chen, Y., Zhang, X.-P., Yuan, J., Cai, B., Wang, X.-L., Wu, X.-L., … Li, X.-Y. (2018, August 21). Data from: Association of body mass index and age with incident diabetes in Chinese adults: a population-based cohort study. Retrieved August 5, 2022, from Datadryad website: https://doi.org/10.5061/dryad.ft8750v

[8] JAEB Center for Health Research. (2019). A Randomized Clinical Trial to Assess the Efficacy in the Management of Type 1 Diabetes. Retrieved August 5, 2022, from JAEB website: https://public.jaeb.org/dataset/563

[9] Michigan Diabetes Research and Training Center. (2012). *High blood sugar (hyperglycemia) - Michigan Medicine*. Adult Diabetes Education Program. Retrieved August 17, 2022, from https://www.med.umich.edu/1libr/MEND/Diabetes-Hyperglycemia.pdf

[10] DataMadness. (2019, October 27). Tensorflow 2 - CPU vs GPU performance comparison. DATAmadness. Retrieved August 14, 2022, from https://datamadness.github.io/TensorFlow2-CPU-vs-GPU