January 24, 2025

Jacob Vierstra

Student ID: 010903152

C964: Computer Science Capstone

Task 2 parts A, B, C and D

Part A: Letter of Transmittal 2

[Part B: Project Proposal Plan 4](#_heading=h.3znysh7)

[Project Summary 4](#_heading=h.2et92p0)

[Data Summary 4](#_heading=h.tyjcwt)

[Implementation 5](#_heading=h.3dy6vkm)

[Evaluation Plan 6](#_heading=h.4d34og8)

[Resources and Costs 7](#_heading=h.2s8eyo1)

[Part C: Application 8](#_heading=h.17dp8vu)

[Part D: Post-implementation Report 9](#_heading=h.3rdcrjn)

[Solution Summary 9](#_heading=h.26in1rg)

[Data Summary 9](#_heading=h.lnxbz9)

[Machine Learning 13](#_heading=h.35nkun2)

[Validation 15](#_heading=h.1ksv4uv)

[Visualizations 15](#_heading=h.44sinio)

[User Guide. 21](#_heading=h.2jxsxqh)

[**Reference Page 25**](#_heading=h.8t1rvg1vc0ph)

Part A: Letter of Transmittal

1/10/2025

Dr Richard Randall

Randall Health Solutions

4001 S 700 E #300, Millcreek, UT 84107

Dear Dr Randall,

We have come to understand that you are overwhelmed by the number of clients needing further assessment for having type 2 diabetes. There are simply too many people potentially having the disease and not enough resources for you to have a complete analysis of each and every individual. This leaves individuals slipping through the cracks when they, in fact, do have diabetes and need intervention. It is so bad, in fact, that "1 in 5 people"(CDC) don't know that they have diabetes. In order to help fix this problem of not having enough resources for doctors and too many clients, we propose a solution using complex machine learning analysis. With basic health markers already achieved from clients' basic tests, you will be able to predict clients who have diabetes without the full analysis. With just the number of pregnancies, glucose, blood pressure, skin thickness, insulin, BMI, diabetes pedigree, and age, we will be able to accurately predict if someone has diabetes. This will benefit you as you will be able to accurately identify which clients have diabetes with common health markers already available. This leads to fewer resources needed by doctors and allows you to focus on clients with more serious health issues. Furthermore, it will increase client satisfaction as they won't be missed for having a disease for months or potentially years. We propose to accurately predict if someone has diabetes with 65% accuracy, which will increase over time with more data.

There are many things to consider when working on a project like this. First is the cost and timeline of the project. Once started, we expect the project to take 3 months with a team of 4 software engineers. This will be a cloud-based service, which will help minimize the cost. Additionally, all the data is already stored in your databases and will have no additional cost. The total estimated cost for this project is $250,000, which is a further breakdown attached below. The other big considerations are data access, maintaining security, and HIPAA compliance. As a result, our access to the database will be limited by cloning the necessary data to a separate database and linking it with a primary key. The primary key will link to their health information so that only those with access to client information can see the individual. As a result, we would only alert if someone is predicted to have diabetes but never has access to personal information. Furthermore, information will be encrypted for added security even though the information will be unable to be linked to anyone.

Lastly, I would like to share my expertise on this topic. I have been a machine learning and AI engineer for 2 years and a software engineer for 4 years. I have various projects that use complex data structures and algorithms to complete various projects ranging from student repositories and complex package routing systems for WGUPS. Furthermore, my personal diabetes pedigree is something of great concern to me, and type 2 diabetes as a whole is a project I deeply desire to aid in solving.

Thank you for taking the time to read this, and I look forward to working with you in the future.

Sincerely,

Jacob Vierstra

Jacob Vierstra, Software Engineer and Project Manager

# Part B: Project Proposal Plan

## Project Summary

The problem is the current need for the medical system to detect type 2 diabetes in patients is failing. Countless people go undetected, which delays treatment and only furthers the disease. Furthermore, the ability of the client to determine if a patient has diabetes is a lengthy and time-consuming process. Randall Health Solutions, or RHS, needs a way to deal with the excess number of patients who need more attention and diagnosis for diabetes. They simply don't have the resources to address this current health crisis. We will provide a solution to this problem through our diabetes prediction project. The deliverables of this project are as follows. The first is to have a functioning prediction software that uses machine learning to predict if someone has diabetes. The second is to be compatible with RHS's current hardware and software. This will be accomplished by implementing a cloud-based solution using Google Collab. Overall, this will greatly benefit RHS as it will reduce the strain on the company and increase customer satisfaction by increasing the speed at which people are diagnosed with diabetes and, as a result, faster medical care and intervention.

## Data Summary

RHS will provide the raw data with their existing client records. This data is available through Kaggle, which provides a good base and maintains HIPAA standards, as there is no personal information attached to the data set. The data will be processed to adjust outliers and missing information. Our goal is to have a prediction rate above 70 percent while also erroring on the side of false positives and not false negatives. If someone is flagged and turns out not to have diabetes, it is not a huge deal, but if someone does have it and isn't detected, it is a much larger issue. With this in mind, our data is processed to eliminate outliers and replace them with the mean values. This allows us to train the model using averages and causes outliers to be more likely to be flagged when predicting. As mentioned before, because of the way the data is acquired, there are no ethical and legal concerns as these are already available publicly. If we had access to a database with client information, this would be a massive concern, but since all our data is anonymous and already public, there are no issues.

.

## Implementation

The industry-standard methodology we are going to use is the waterfall methodology. This project has clear outcomes and deliverables that aren't going to change, so there is no need to have a more flexible methodology like agile. We already have a clear set of tasks that need to be completed and know how to accomplish them effectively. Additionally, until the model is trained and functional, there is nothing that is functional to RHS. As a result, the typical disadvantage of waterfall having a delayed delivery isn't an issue, as training the model is one of the last steps. Below are the waterfall stages:

* Requirements: In this stage, we will work with RHS to ensure requirements and timelines are established and clear between both parties.
* Design: In this stage, we will determine the exact deliverables and create an outline of how we are going to accomplish them.
* Implementation: In this stage, we will develop the product and accomplish the deliverables and requirements agreed upon.
* Verification: In this stage, we will verify the product functions properly and test edge cases to ensure they function as expected.
* Maintenance: In this stage, we will ensure the continued functionality of our product and have continued support as needed.

Below is the projected timeline of the project. It is estimated to take approximately 95 hours to complete, with a projected completion date of January 10, 2025.

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone or deliverable | Duration  (hours or days) | Projected start date | Anticipated end date |
| Requirement analysis | 10 hours | 12/15/24 | 12/18/24 |
| Designing the architecture | 10 hours | 12/19/24 | 12/23/24 |
| Development and implementation | 45 hours | 12/26/24 | 1/3/25 |
| Verification and testing | 15 hours | 1/4/25 | 1/7/25 |
| Deployment | 15 hours | 1/8/25 | 1/10/25 |

## Evaluation Plan

Verification and validation are essential to ensure outcomes and expectations are managed and met. With the waterfall method, there is a dedicated verification stage. In this stage, we will ensure the product is functioning properly, test edge cases, and ensure that the metrics set in previous stages are met. Additionally, we will ensure the data is as expected since it is possible to have a seemingly functional project while having inaccurate or wrong data. Validation results are as expected, ensuring the program not only functions but functions properly. Lastly, in each stage of development, there will be verification before moving on to the next stage. The verification methods to be used at each stage of development are as follows:

* The requirements stage will be evaluated and verified by having a written agreement on the deliverables and goals of the project.
* The design stage will be evaluated and verified by ensuring it will be able to accomplish the deliverables and goals of the project with a clear outline of how this will be accomplished.
* The implementation stage will be evaluated and verified by ensuring the project is developed with deliverables and goals in mind and that they are achieved.
* The verification stage will be verified by ensuring everything is functioning as expected, even in edge cases, and ensuring there are no bugs in the project. If there are common bugs, they will be addressed and fixed at this stage.
* The maintenance stage will be verified by ensuring the functionality of the project continues over time. This should be something that requires little maintenance, but if there are new health metrics that the customer wants added after using the project, it would be done here.

## Resources and Costs

|  |  |  |
| --- | --- | --- |
| Resource | Description | Cost |
| Employees | This project will take approximately four software engineers 3 months to complete, with an average salary of $80,000 per year. | $80,000 |
| Hardware | There are no additional hardware costs as the current computers are already sufficient to run the database, and the rest is on a web application. | $0 |
| Third-party cost | A | $5,000 |
| Continued support | Continued support is recommended for up-keep, troubleshooting, bug fixing, and program improvement. This price is reflected in a year of support. | $20,000 |
|  | Total | $105,000 |

# Part C: Application

# The list of downloaded files for this project are:

* C964\_Capstone\_Vierstra.ipyn
* diabetes.csv

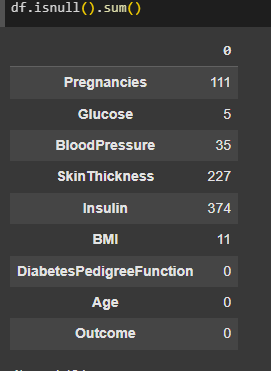
# Part D: Post-implementation Report

## Solution Summary

The problem was that the medical system's previous need to detect type 2 diabetes in patients was failing. Countless people had gone undetected, which delayed their treatment for the disease. Furthermore, the ability of the client to determine if a patient had diabetes was a lengthy and time-consuming process. We solved this problem through our diabetes prediction project. The deliverables of this project were achieved as follows. First, we have a functioning prediction software that uses machine learning to predict if someone has diabetes. By simply putting in your client's health data, we can predict whether or not you likely have diabetes and trigger further testing. Second, it is current hardware and software. We used Google Collab and imported the data into a CSV file to run the software. All computers at RHS are able to run this software as it is web-based and uses a standard CSV file. This has greatly benefited RHS as it has reduced the strain on the company's employees. Additionally, it has increased customer satisfaction by increasing the speed at which people are diagnosed with diabetes.

## Data Summary

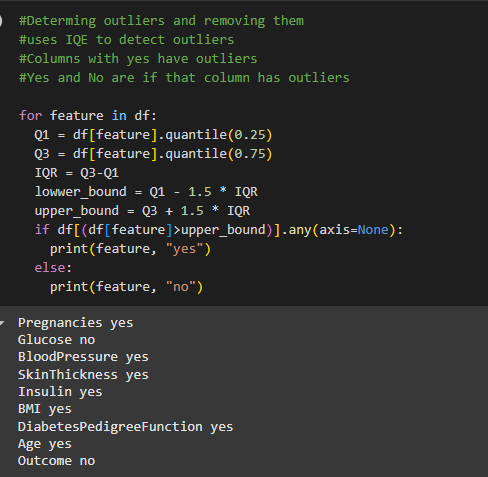
The raw data is from Kaggle.com. It contains various health metrics, as well as determining whether the person has diabetes or not. The data was processed with various things in mind. In the design stage, we decided we wanted to have the model be biased to give false positives. If someone has a false positive and receives further invention from a doctor only to find out they don't have diabetes, it is no big deal. However, if someone receives a false negative and doesn't know to go get treatment, it is a major concern with serious moral and ethical implications. In order to account for this, we decided to adjust it by preprocessing the data. This leads to the development stage, where we process the data. First, there were various null values in the project. As seen below,

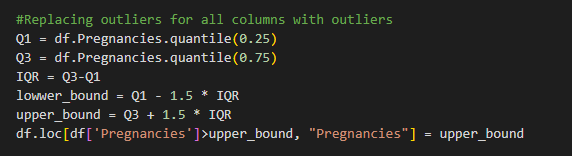


By using the isnull() function, we see that there are many missing values. We did not want this to affect the training of the model, so we filled the null value with each for that given outcome and data point. Using the groupby() and agg() functions, we are able to find the mean for each group.



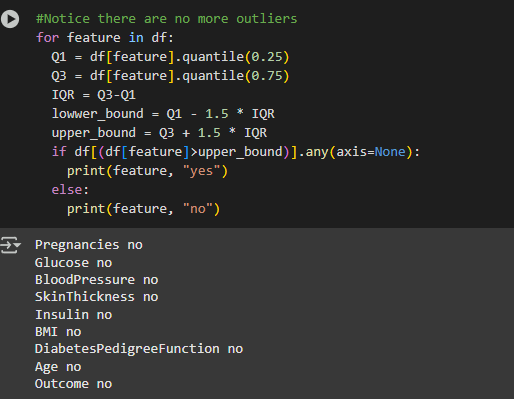


For instance, we can see that the mean of insulin without diabetes is 109.98, and with diabetes is 141.25. If there is a null value for someone's insulin, we replace it with the mean value. By replacing it with the mean, we were able to train the model while minimizing the impact of null values. Additionally, in the development stage, we eliminated variables that were outliers. This further biases to false positives of predicting for patience. By being trained without outliers, when someone inputs their data to get a prediction, it is more likely to flag for diabetes. This hurts the accuracy of the model but is an intentional decision due to the ethical and moral concerns stated above. 

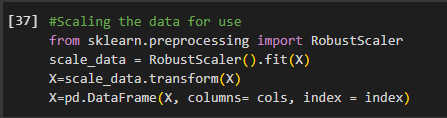
As shown above, each column with a yes indicates that there were outliers that had to be replaced. This means that for seven columns, we further processed the data for those outliers. We replaced the outliers with the upper bound for each column, respectively. This is done by calculating the interquartile range (IQR) to find the upper bound and replace values outside of it. 

Above is the code used to do so for a particular column, and it was repeated for all columns with outliers.

After completing that processing, we re-ran the same code as before to show that there are no more outliers, as seen below.



Lastly, we scaled the data to be used in order to optimize the algorithm. It also allows for ease of implementation of prediction data and enables easy seeing of the impact of different health markers.



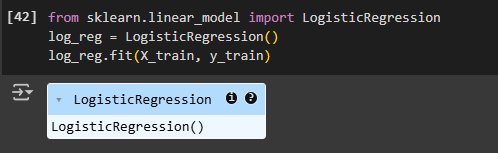
Above is the scaling method on the data set, which we will be able to use later on for user input data. Once our data has been scaled, we can know the impact of the health markers based on being a positive or negative value.



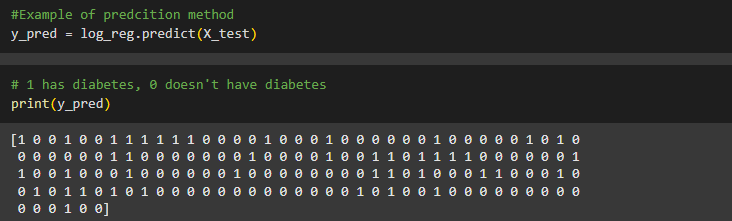
Above is the head of our data set after being scaled. A negative number contributes to not having diabetes, while a positive number contributes to having diabetes.

## Machine Learning

The method used to create a supervised machine learning algorithm model for this project was logistic regression. This method determines the odds of something being true, in our case, having diabetes or not, by using a linear combination of one or more variables. In our case, we used a linear combination of 8 health markers variables to determine if someone had diabetes. The reason for choosing this method was it was a simple model to implement while also easily accounting for all the variables we had. We simply scaled the data (as shown in the data summary), added the training functionality, and inserted it into the logistic regression algorithm. For instance, below is the implementation of the logistic regression algorithm.



With this trained algorithm, we are then able to predict whether or not someone will likely have diabetes.

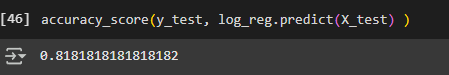


As shown above, by printing the prediction method, we are able to see what the model predicts for each set of inputs.

## 

## Validation

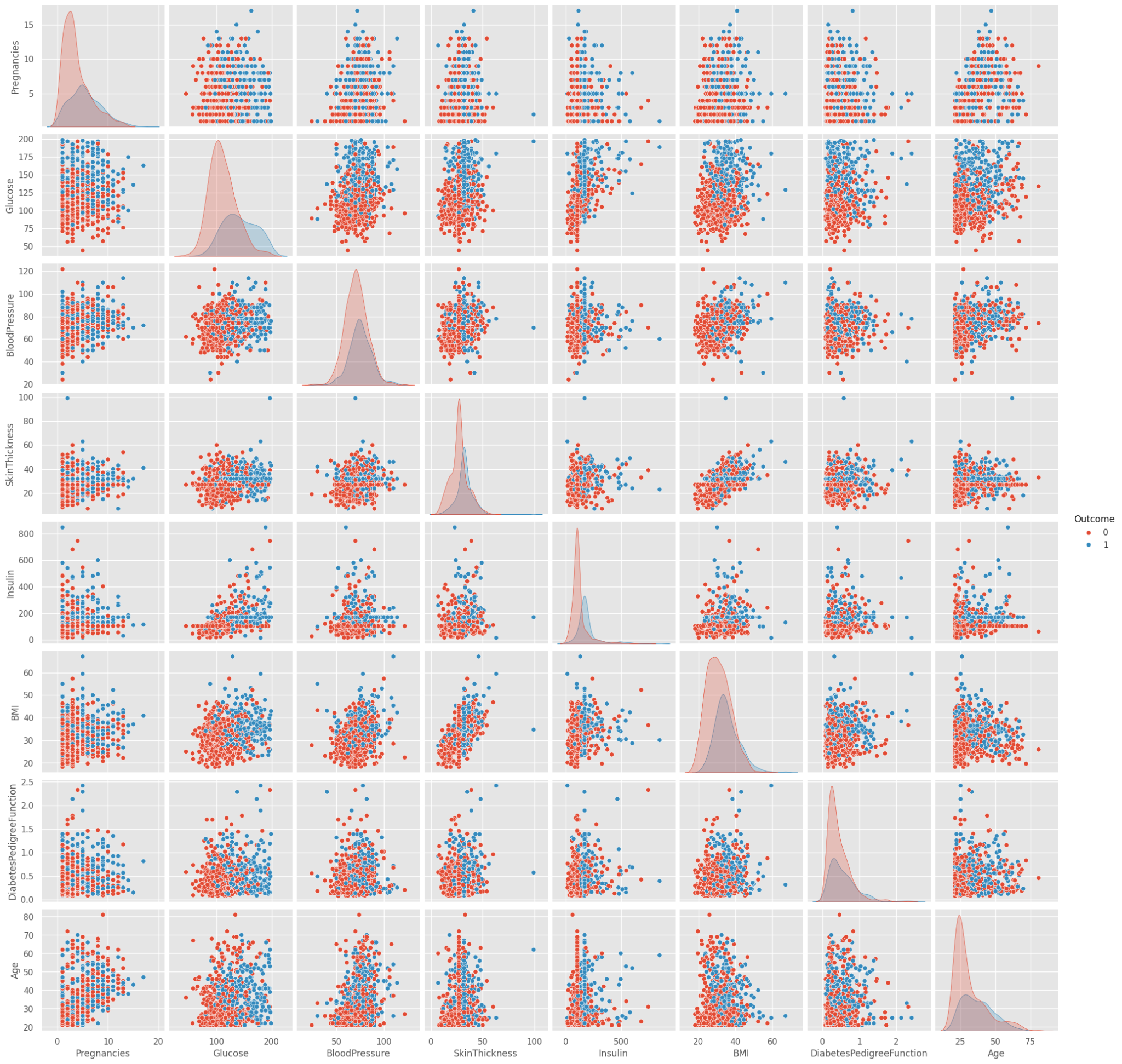
For our employed method of logistic regression, we decided to validate the results using the accuracy method. Accuracy is calculated by taking the number of correct predictions divided by the total predictions.



As shown above, our overall accuracy was 81.81%, which exceeded our initial goal of above 60%.

## Visualizations

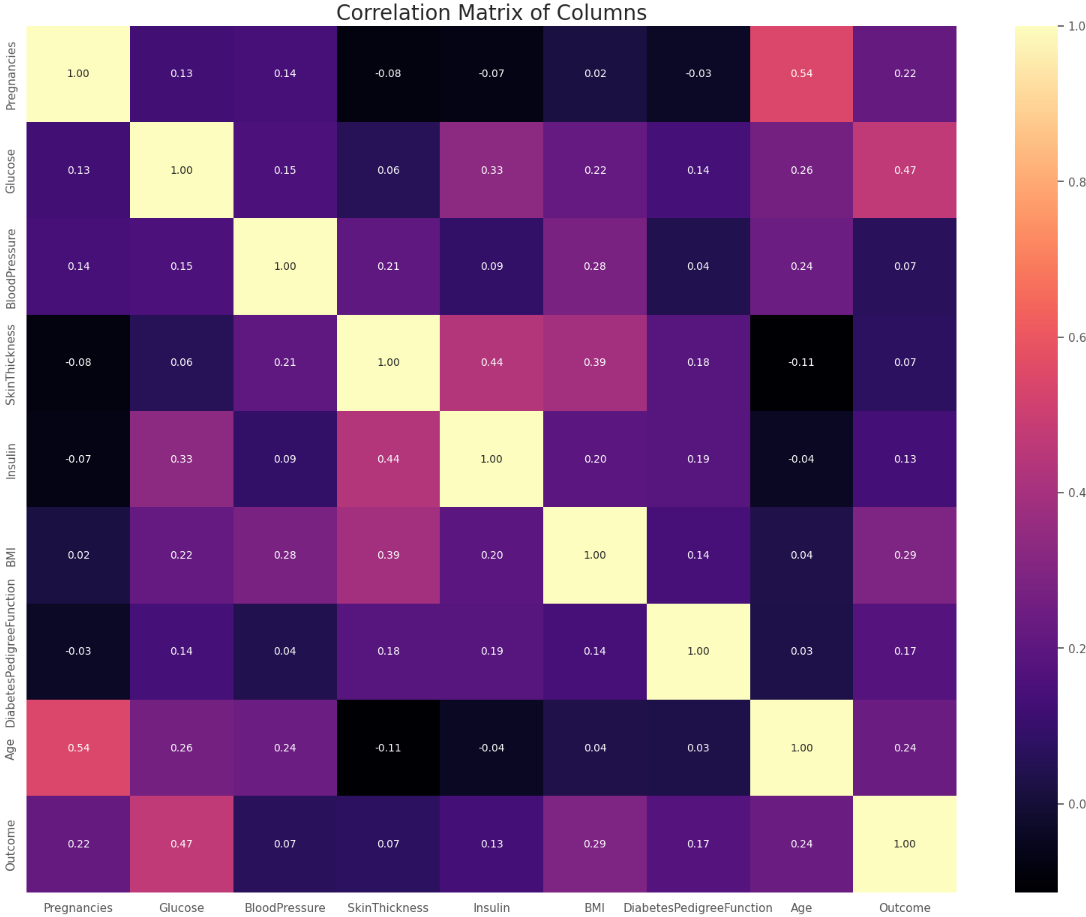
Visualization of the data has revealed many insights about having diabetes. Although there are many contributing factors to diabetes, we are able to see which have the most impact.

First is the pair plot of all the variables. The graph shows the value of each variable on an x-axis and y-axis to its respective variable. It is color-coded, so the red dots represent people without diabetes, and the blue has diabetes. The most striking correlation is that of glucose and BMI. You can clearly see that higher glucose and a higher BMI strongly correlate to having diabetes. This is very helpful to see but also hard to quantify.

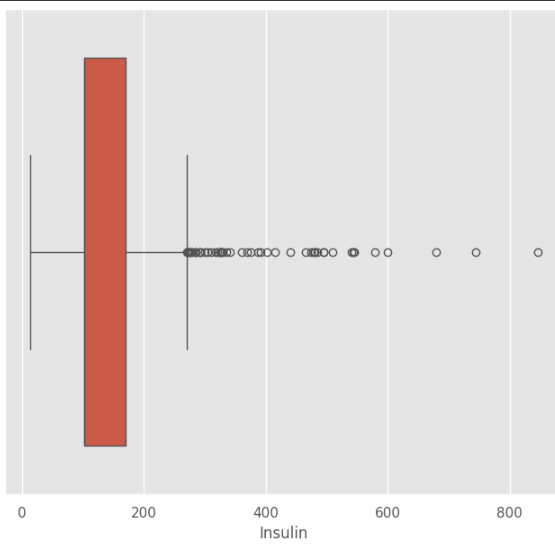


Here, the BMI (y-axis) and glucose (x-axis) are enlarged.

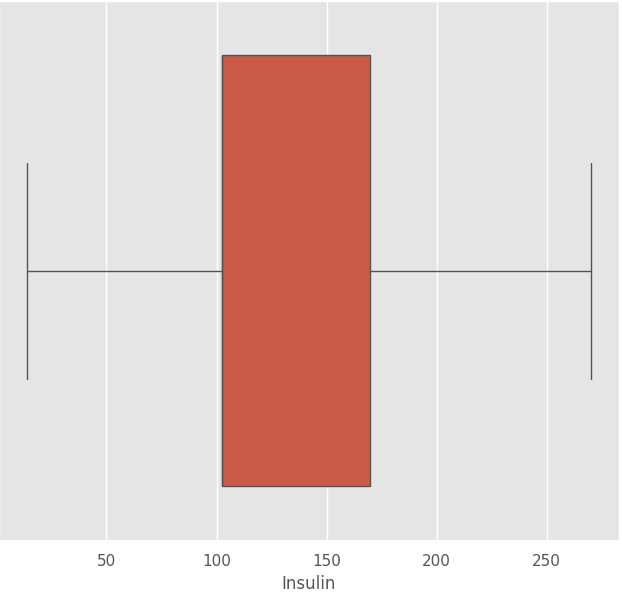
With the next visualization, we are not only able to see this correlation but also have quantitative data.



This is a correlation matrix of all the values given. The higher the number, the stronger the correlation it has with the corresponding health marker. The value of 1.0 is just when the x and y of the matrix are the same marker. Looking at the outcome column, we can see that glucose (0.47), BMI (0.29), and age (0.24) have the strongest correlation for having diabetes. Another insight that can be gained from visualizing what the outliers are is



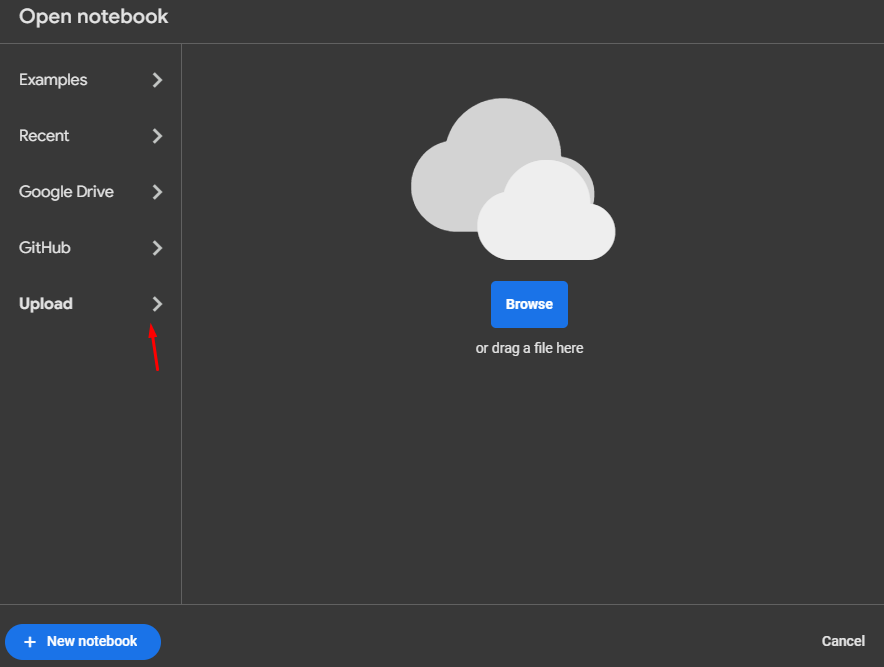
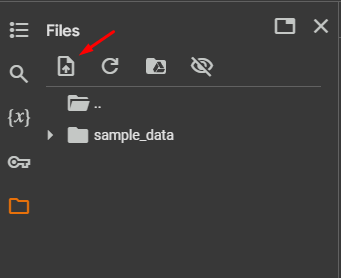
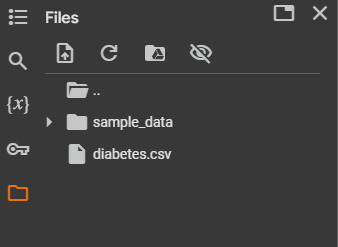
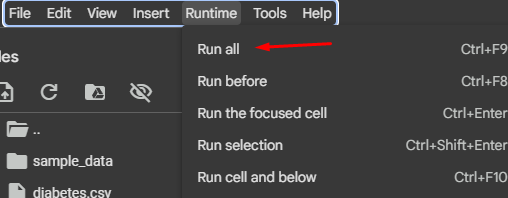
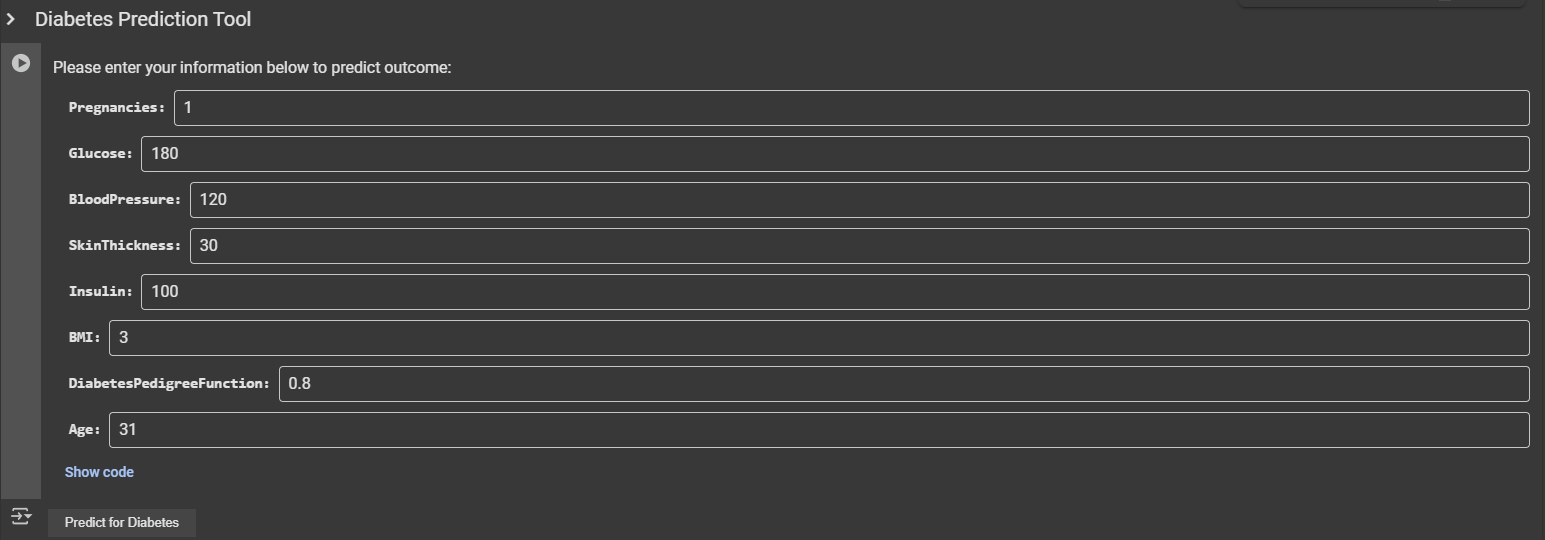
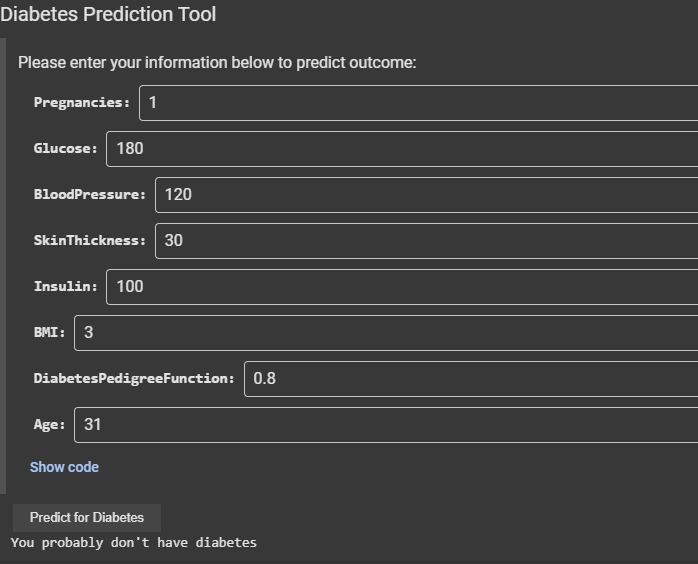
Here, we can see that the most values for insulin are within the red rectangle and the interquartile range by the two vertical lines at approximately 20 and 270. Anything outside of the interquartile range is considered an outlier and something that we filtered out of the data.

Below is the same graph function for insulin after the outliers have been replaced. 

## 

## User Guide.

This project all takes place using Google Collab. Here are the steps to follow to get the application up and running.

1. Download all attached files.
2. Log into your Google account and open Google Collab
3. When prompted to open the notebook, select upload and choose the downloaded file "C964\_Capstone\_Vierstra.ipyn"
   1. 
4. Load the database into the notebook.
   1. Click the files tab on the left-hand side, click upload and choose "diabetes.csv"
   2. The file should now appear under “sample\_data” as shown: 
5. Navigate to Runtime in the toolbar, click "Run all" and wait a few minutes for it to complete
6. To use the UI to make a diabetes prediction, scroll to the bottom cell titled "Diabetes Prediction Tool" and enter your information. 
7. Once all your information is imputed, click on the button "Predict for Diabetes" to see the resulting output below.

# 

# Reference Page

**Centers for Disease Control and Prevention. (n.d.).** Diabetes statistics. *U.S. Department of Health and Human Services.* Retrieved January 24, 2025, from<https://www.cdc.gov/diabetes/communication-resources/diabetes-statistics.html>

**Mathchi. (n.d.).** Diabetes data set [Data set]. *Kaggle.* Retrieved January 24, 2025, from<https://www.kaggle.com/datasets/mathchi/diabetes-data-set/data>