May 24, 2024

C964: Computer Science Capstone

Task 2 parts A, B, C and D

Part A: Letter of Transmittal 1
Part B: Project Proposal Plan 3
- Project Summary 3
- Data Summary 4
- Implementation 5
- Timeline 5
- Evaluation Plan 5
- Resources and Costs 7

Part C: Application 7
Part D: Post-implementation Report 7
- Solution Summary 7
- Data Summary 8
- Machine Learning 9
- Validation 10
- Visualizations 11
- User Guide 14

Reference Page 16

Part A: Letter of Transmittal

Kevin Colagiovanni

2177 Weston Pl

Santa Clara, CA 95054

24-May-2024

Dr. Linda Carter

HillCrest Diabetes Center

624 W Fremont Blvd, Suite 24

Fremont, CA 94538

Dear Dr. Linda Carter,

Diabetes is on the rise, and doctors and their staff need help diagnosing diabetes quickly and accurately. Diabetes is largely preventable and can be avoided by making lifestyle changes. These changes can also lower the chances of developing heart disease and cancer. So, there is a dire need for a prognosis tool that can help the doctors with early detection of the disease and hence can recommend the lifestyle changes required to stop the progression of the deadly disease.

We are proposing an application that is designed to be used in an endocrinologist's office to help diagnose diabetes faster and more accurately than the current method. It can be used to aid the current method or, because of its accuracy, it can be used to replace the current method.

The application that we are proposing will benefit your practice by helping to diagnose diabetes faster and more accurately than the current method, making it possible to have the capacity to help more patients per day, bring in more money to the clinic, use cutting edge technology to save and improve lives every day, and save the you and your staff time and having one less thing to worry about so your time cna be better spent helping patients who need your time in ways other than diagnosing diabetes.

If approved and implemented, the application development and deployment would cost around $7,500 and could be designed and deployed in less than two weeks depending on configuration and options. There would be no other cost associated and we would support the application for three years, which includes software patches and upgrades.

The proposed application would ask for the patients blood glucose levels, blood pressure, BMI, insulin levels, number of pregnancies, diabetes pedigree function, skin thickness, and age, then it would use a machine learning algorithm that has been trained using a highly accurate and large dataset and make a prediction as to whether the patient has diabetes or not.

There are not any ethical or legal considerations or precautions that would need to be considered when working with and communicating about sensitive data because the dataset used is a public dataset.

I personally have over 20 years of experience in the biotech industry and 8 of those have been with our company, developing and deploying software solutions like this one. We are confident that we can help your clinic quickly and accurately predict diabetes in your patients. Saving your clinic time and money.

I have enclosed a more detailed description about the proposed application for your review. I am happy to discuss any questions or concerns that you may have. Thank you for your time and I hope to hear from you soon.

Sincerely,

Kevin Colagiovanni

Kevin Colagiovanni, CTO

Part B: Project Proposal Plan

Project Summary

The problem is that diabetes is on the rise, and doctors and their staff need help diagnosing diabetes quickly and accurately. Diabetes is largely preventable and can be avoided by making lifestyle changes. These changes can also lower the chances of developing heart disease and cancer. So, there is a dire need for a prognosis tool that can help the doctors with early detection of the disease and hence can recommend the lifestyle changes required to stop the progression of the deadly disease. (Deep learning approach for diabetes prediction using PIMA Indian dataset - Huma Naz[obj] and Sachin Ahuja)

This application is designed to be used in an endocrinologist's office to help diagnose diabetes faster and more accurately than the current method. It can be used to aid the current method or because it is very accurate, it can be used to replace the current method.

Application Deliverables:

- The application, which will be compatible with Windows, MacOS, and Linux operating systems.
- A user guide explaining how to install and use the application.
- Full setup and verification of functionality of the application in the clinic by an experienced staff member as well as in person training.
- 3 years of product support and updates of the application.
- [Optional] A laptop PC with the application installed and ready to use.

This application will benefit the client by helping the clinic that it is implemented in to diagnose diabetes faster and more accurately than the current method making it possible to have the capacity to help more patients.

Data Summary

The source of the data for the proposed project is Kaggle.com (https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database). The method for the data collection will be through direct medical examinations and tests administered by health professionals. This would typically involve measuring blood glucose levels, blood pressure, BMI, and other relevant health indicators through standard medical procedures.

An advantage of the data collection method described is the data collected is highly relevant to diabetes prediction, which enhances the quality of the data.

A limitation of the data collection method described is that the dataset contains missing values for some attributes, which can complicate data analysis and model training which can introduce additional complexity and potential bias.

The data will be processed and managed throughout the application development lifecycle by:

- Designing a development plan that involves using the dataset efficiently and deciding on the best machine learning algorithm to use for predictions.
- Developing an application that relies on the dataset to function quickly and accurately when applying machine learning to it for making predictions.
- It will be maintained by checking for updates to the dataset (ie. new data added) periodically. If a better more efficient machine learning algorithm is identified, then it will be implemented.

There are technically no missing values in the dataset, but in actuality in this particular dataset all the missing values were given 0 as a value which is not good for the authenticity of the dataset. The data will be prepared for use by the machine learning algorithm for the application by replacing all 0 values with NAN values and then replacing the NAN values with the mean value of that specific column. The following shows the percentage of 0 values in the dataset:

- Pregnancies - 0% (There are 0 values in this column, but it means the patient has never given birth)
- Glucose - 0.7%
- Blood Pressure - 4.6%
- Skin Thickness - 29.6%
- Insulin - 48.7%
- BMI - 1.4%
- Degree Pedigree Function - 0%
- Age - 0%

There are no special behaviors that should be exercised when working with and communicating about sensitive data in the development of the application because there is no sensitive data in the dataset, it is a public dataset.

Implementation

SEMMA will be the standard methodology that will be applied to the implementation of my proposed project by doing the following:
Sample: This optional stage is not being applied to this project.
Explore the data by using MatPlotLib's *plot.bar, plot.scatter,* and *plot.hist* to visualize it.
Modify the dataset by replacing 0 values in each column, except the "Pregnancies" and target columns, where there should be actual data. This will be done using Pandas' *replace* method (0, NumPy.NaN) which will replace 0 values with Nan values, then using Pandas' *fillna* method to replace Nan values with the average value of the given column.
Model the dataset using the sci-kit learn's *DecisionTreeClassifier* machine learning model.
Assess the data by checking the accuracy of the training using sci-kit learn's *accuracy* method.
(KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW - Ana Azevedo and M.F. Santos)

Timeline

Milestone	Hours	Start Date	End Date	Working Days
Planning and design	6	28-May-2024	28-May-2024	1
Development	20	29-May-2024	31-May-2024	3
Documentation	10	03-June-2024	04-June-2024	2
Total	36	28-May-2024	04-June-2024	6

Evaluation Plan

The application will be validated throughout each stage of development by doing the following:

- Gather the business requirements for validation testing from the end user.
- Prepare the business plan and send it for approval to the onsite/stakeholders involved.
- On approval of the plan, begin to write the necessary test cases and send them for approval.
- Once the business plan is approved, complete testing with the required software and environment will begin and send the deliverables as requested by the client.
- Upon approval of the deliverables, UAT testing will be performed by the client.
- Then software will go into production.

  (Validation Testing Ultimate Guide - Sruthy)

Once the development of the application is complete, it will be verified by doing the following:

- *Peer Reviews:* The easiest method and most informal way of reviewing the documents or the programs/software to find out the faults during the verification process is the Peer-Review method. In this method, we give the document or software programs to others and ask them to review those documents or software programs where we expect their views about the quality of our product and also expect them to find the faults in the program/document. The activities that are involved in this method may include SRS document verification, SDD verification, and program verification. In this method, the reviewers may also prepare a short report on their observations or findings, etc.

- *Walk-Through:* Walk-throughs are a formal and very systematic type of verification method as compared to peer review. In a walkthrough, the author of the software document presents the document to other persons which can range from 2 to 7. Participants are not expected to prepare anything. The presenter is responsible for preparing the meeting. The document(s) is/are distributed to all participants. At the time of the meeting of the walk-through, the author introduces the content in order to make them familiar with it and all the participants are free to ask their doubts.

- *Inspections:* Inspections are the most structured and formal type of verification method and are commonly known as inspections. A team of three to six participants is constituted which is led by an impartial moderator. Every person in the group participates openly, and actively, and follows the rules about how such a review is to be conducted. Everyone may get time to express their views, potential faults, and critical areas. After the meeting, a final report is prepared after incorporating necessary suggestions from the moderator.

  (Verification Methods in Software Verification - Geeks for Geeks)

Resources and Costs

The resources and all associated costs needed to implement the proposed solution would be:

- PC to run the program - $450
- Linux Ubuntu OS installation and setup (2hrs x $100) - $200
- Develop and package the App (36hrs x $150/hr) - $5400
- Blood Glucose Analyzer - $1100

Additional costs per patient:

- Third party service to perform the patients blood work(insulin) - $500
- Medical appointment for other medical measurements (calculate BMI, measure skin thickness, take blood pressure), and other non-medical data needed (age, determine degree pedigree function, number of pregnancies) - $800

Part C: Application

The application is provided in the “C964_CS_Capstone_Project.zip” file. The contents of the zip file are:

- main.py - The main window, where the application starts. This is the file that needs to be run to begin the application.
- Plot_data.py - The plot window and the plots.
- Process_and_train_data.py - The prediction window and the training and predictions.
- Diabetes.csv - The dataset

Part D: Post-implementation Report

Solution Summary

The problem is that diabetes is on the rise, and doctors and their staff will need help diagnosing diabetes quickly and accurately. This application proposes to use machine learning to help doctors quickly and accurately detect diabetes in patients based on a number of data points. With input from doctors and some code adjustment, it could also be used for early detection of diabetes. The application will provide the solution to this problem by using the patient’s medical data as input data entered by medical staff and output if the patient has diabetes or not. It uses a diabetes dataset consisting of 768 rows, with 8 input values(columns) and one target value(column) which is either true(1) indicating that the patient does have diabetes, or false(0) indicating that the patient does not have diabetes. There is also functionality to graph the input data to help visualize the dataset.

Data Summary

The source of the data for the proposed project is Kaggle.com (https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database)

The method for the data collection was through direct medical examinations and tests administered by health professionals. This would typically involve measuring blood glucose levels, blood pressure, BMI, and other relevant health indicators through standard medical procedures.

There are technically no missing values in the dataset but that is actually not a true story as in this particular dataset all the missing values were given 0 as a value which is not good for the authenticity of the dataset. The data will be prepared for use by the machine learning algorithm from part C2 for my proposed project by replacing all 0 values with NAN values and then replacing the NAN values with the mean value of that specific column. The following shows the percentage of 0 values in the dataset:

- Pregnancies - 0% (There are 0 values in this column, but it means the patient has never given birth)
- Glucose - 0.7%
- Blood Pressure - 4.6%
- Skin Thickness - 29.6%
- Insulin - 48.7%
- BMI - 1.4%
- Degree Pedigree Function - 0%
- Age - 0%

The Diabetes Dataset:

```
Python 3.11.0 (main, Apr 19 2024, 14:55:57) [GCC 7.5.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas as pd
>>> diabetes_dataset = pd.read_csv('diabetes.csv')
>>> print(diabetes_dataset)
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  DiabetesPedigreeFunction  Age  Outcome
0              6      148             72             35        0  33.6                     0.627   50        1
1              1       85             66             29        0  26.6                     0.351   31        0
2              8      183             64              0        0  23.3                     0.672   32        1
3              1       89             66             23       94  28.1                     0.167   21        0
4              0      137             40             35      168  43.1                     2.288   33        1
..           ...      ...            ...            ...      ...   ...                       ...  ...      ...
763           10      101             76             48      180  32.9                     0.171   63        0
764            2      122             70             27        0  36.8                     0.340   27        0
765            5      121             72             23      112  26.2                     0.245   30        0
766            1      126             60              0        0  30.1                     0.349   47        1
767            1       93             70             31        0  30.4                     0.315   23        0

[768 rows x 9 columns]
>>> print(diabetes_dataset.describe())
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin         BMI  DiabetesPedigreeFunction         Age     Outcome
count   768.000000  768.000000     768.000000     768.000000  768.000000  768.000000                768.000000  768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458   79.799479   31.992578                  0.471876   33.240885    0.348958
std       3.369578   31.972618      19.355807      15.952218  115.244002    7.884160                  0.331329   11.760232    0.476951
min       0.000000    0.000000       0.000000       0.000000    0.000000    0.000000                  0.078000   21.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000   27.300000                  0.243750   24.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000   32.000000                  0.372500   29.000000    0.000000
75%       6.000000  140.250000      80.000000      32.000000  127.250000   36.600000                  0.626250   41.000000    1.000000
max      17.000000  199.000000     122.000000      99.000000  846.000000   67.100000                  2.420000   81.000000    1.000000
>>>
```

Machine Learning

The application uses a dataset from the National Institute of Diabetes and Digestive and Kidney Diseases that has 8 different medical variables and one target variable. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset(Pima Indians Diabetes Database - UCI MACHINE LEARNING and Kaggle Team).

A supervised learning classification algorithm was chosen to make a prediction because the target was categorical and the target values were known. Decision tree classification is a Random Forest machine learning algorithm that uses multiple decision trees to improve classification and prevent overfitting(Supervised Machine Learning - Geeks for Geeks).
A decision tree is a flowchart-like tree structure where each internal node denotes the feature, branches denote the rules and the leaf nodes denote the result of the algorithm. It is a versatile supervised machine-learning algorithm. It is a Random Forest algorithm used to train different subsets of training data, which makes random forest one of the most powerful algorithms in machine learning(Decision Tree - Geeks for Geeks).

The *pandas* and *numpy* libraries were used to prepare the dataset. The *sci-kit learn* library was used to split the data, train the model using the training data, check the accuracy using the test data, and predict the outcome. The *joblib* library was used to save and load a persisting model, to save time when making a prediction.

The application will predict if a patient has diabetes using 8 different medical variables. This happens because a machine learning algorithm was implemented. A csv dataset was read into a pandas dataframe, that dataframe was then randomly split into two parts using sci-kit learn's *train_test_split* method. Eighty percent of the data frame was used to train the model using sci-kit learn's *fit* method, and the remaining twenty percent was used for testing. A prediction is made with the test data using sci-kit learn's *predict* method. The accuracy of the training was tested using sci-kit learn's *accuracy_score* method. The application gives the user the option to use the original dataset, unmodified, or to use a cleaned version. The dataset has a lot of 0 values, which is okay in the pregnancies and target columns, but in all other columns it means that the value is invalid and lowers the accuracy of training and ultimately the predictions. The cleaned version of the data converts the 0 values in all columns except the pregnancies and target columns to the average value of the column which can then be used for training, which may increase the accuracy of the training model and ultimately the predictions.

Supervised learning was selected because the dataset being used has labeled columns and because it has input and output parameters. Classification is the type of supervised learning that was selected because the outcome or target can be classified or categorized into two different categories. The supervised machine learning algorithm that was selected as the best choice was the Decision Tree Classification algorithm. The decision tree classification algorithm was selected because it is used to model decisions and their possible consequences. Each internal

node in the tree represents a decision, while each leaf node represents a possible outcome. Decision trees can be used to model complex relationships between input features and output variables.In the training process, the data is split 80/20. 80% as training data and the rest as testing data. The model learns from training data only. Learning means that the model will build some logic of its own. Once the model is ready it can be tested. At the time of testing, the input is fed from the remaining 20% of data that the model has never seen before, the model will predict some value and will compare it with the actual output and calculate the accuracy. (Supervised Machine Learning - Geeks for Geeks).

Validation

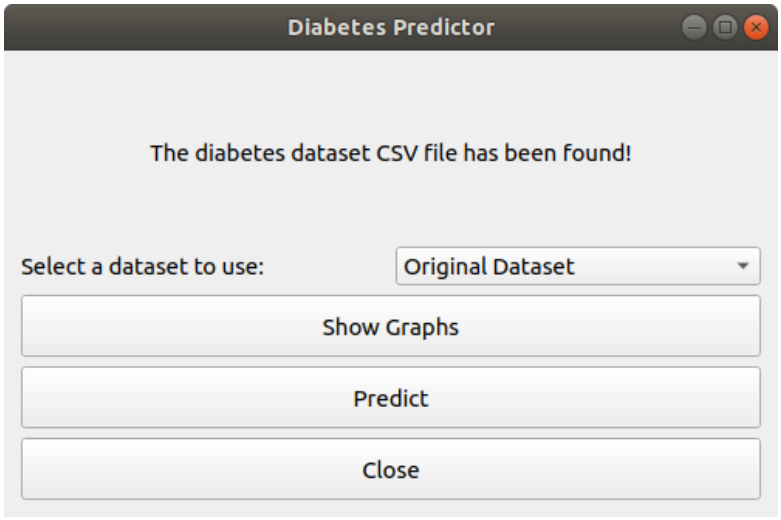
The accuracy of the machine learning application was accessed using Sci-Kit Learn's metric module *accuracy_score* method. It takes the Y values of the test data and the prediction as inputs and outputs the accuracy score, which is a value between 0 and 1. Multiplying this number by 100 will give the accuracy percent. Classification accuracy can be described as the "percentage of true prediction" or it is a sum of the true positive and true negative divided by the sum of predicted class value, it can be calculated using the following formula:

$$X = t / n * 100$$

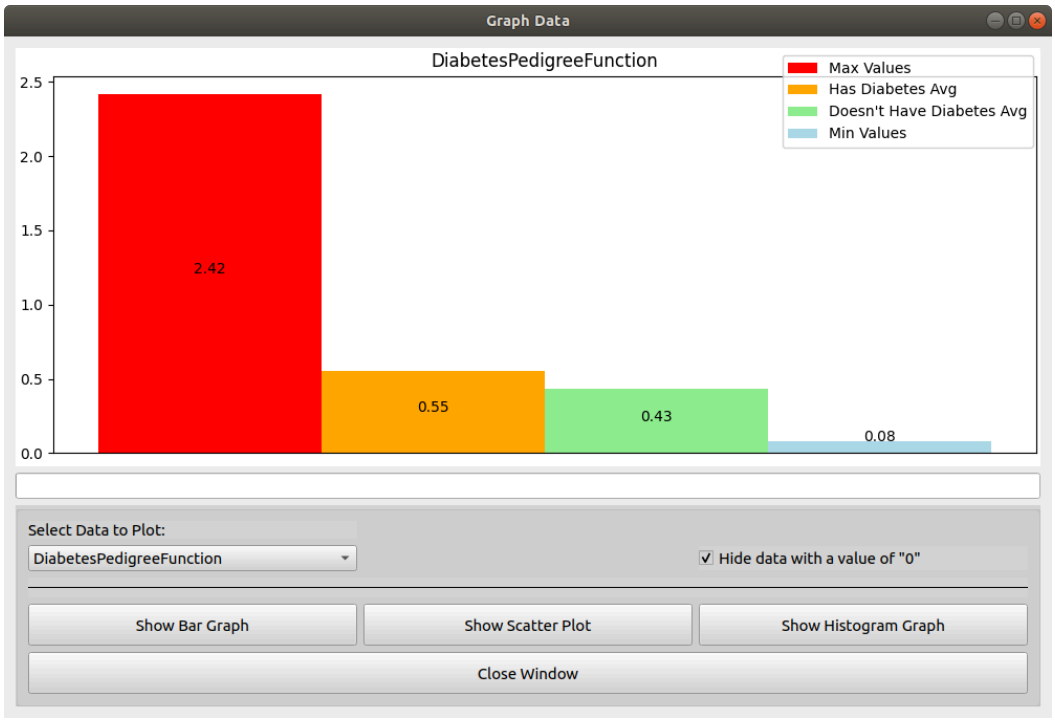
Here X represents the classification accuracy, t is the number of correct classification and n is a total number of samples. (Deep learning approach for diabetes prediction using PIMA Indian dataset - Huma Naz[OBJ] and Sachin Ahuja)

Visualizations

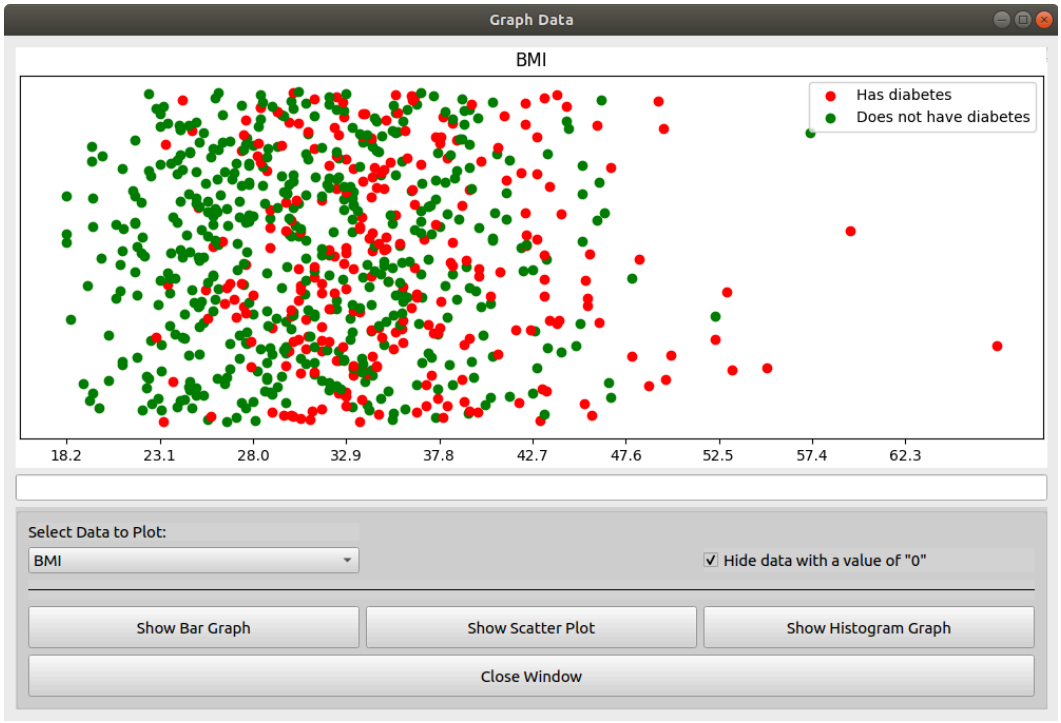
Main Window:



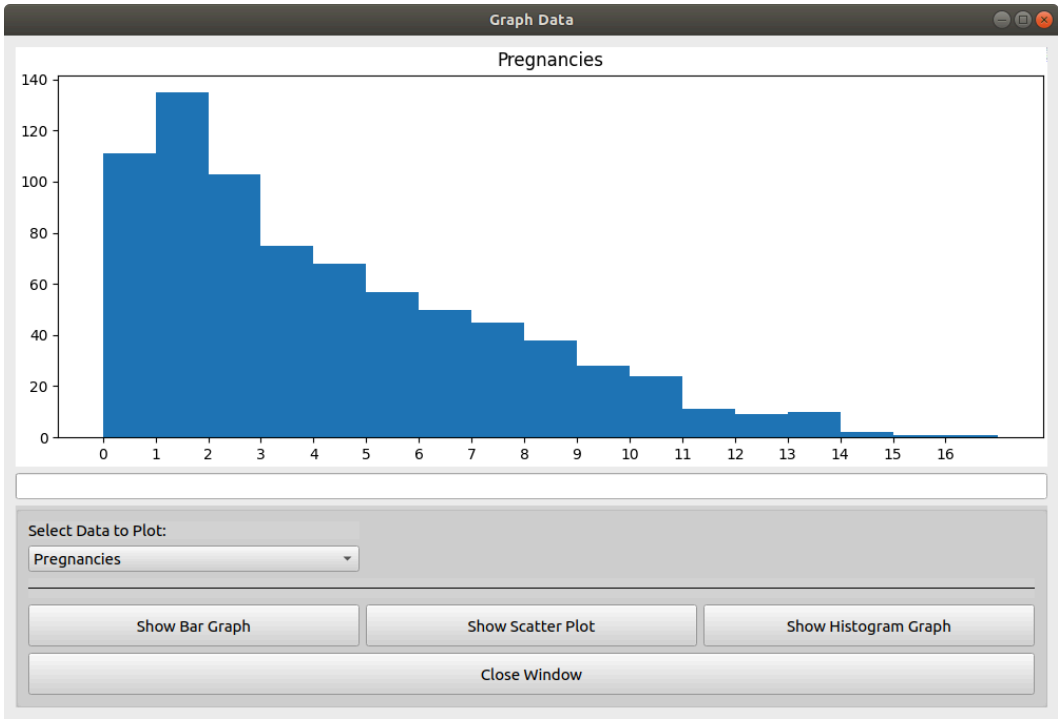
Bar Graph:



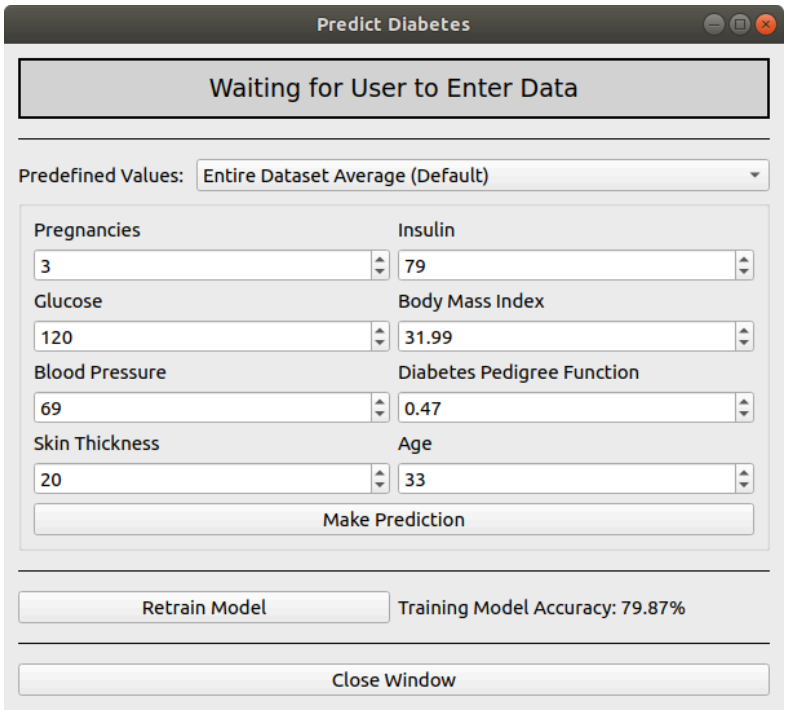
The Scatter Plot:



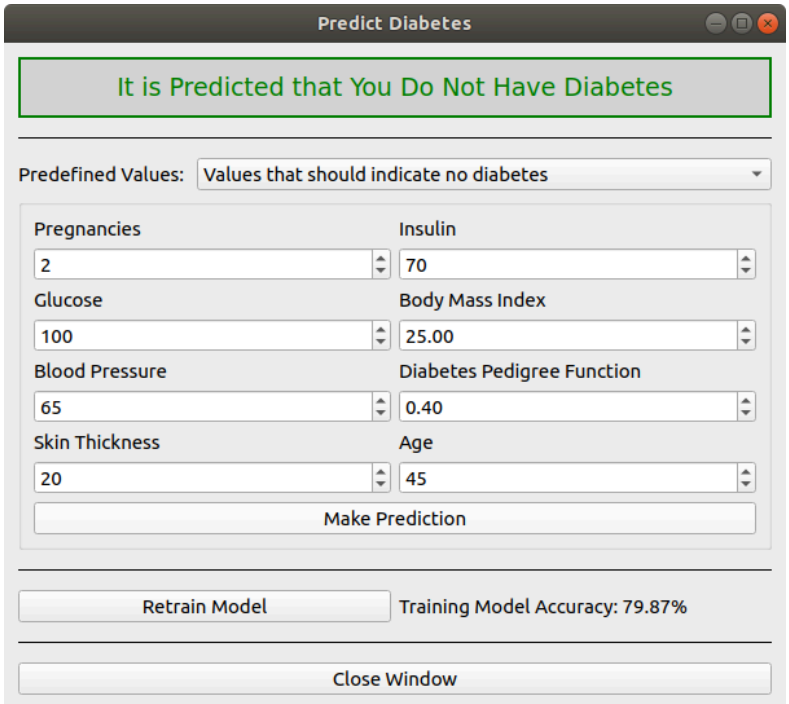
The Histogram Graph:



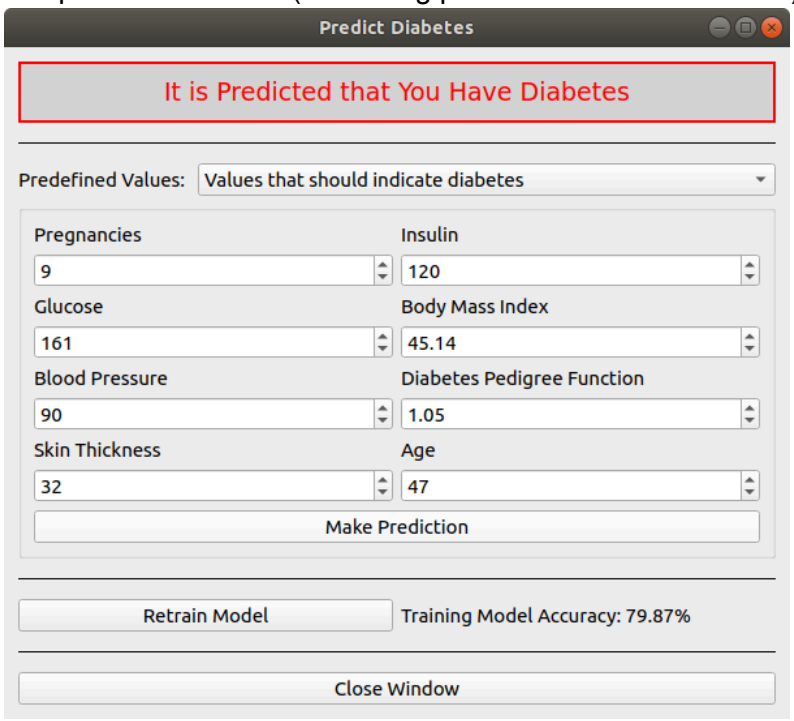
The Prediction Window:



The prediction window(Predicting patient does not have diabetes):



The prediction window(Predicting patient does have diabetes):



User Guide

Instructions for downloading and installing necessary software and libraries.

Note: The following instructions are intended for a PC running Windows 10 or higher.

1. If it's not already installed on the PC, download Python 3.11 from https://www.python.org/downloads/release/python-3119/(The application may work with Python versions 3.8 through 3.10 or 3.12, but it was built, tested, and only guaranteed to work correctly using Python version 3.11)
2. Install Python using the instructions linked below, after a successful installation, continue to the next step: https://www.geeksforgeeks.org/how-to-install-python-on-windows/
   Note: A command prompt window can be opened by pressing the windows key and typing "cmd".
3. In a command prompt window type: "python –version" or "python3 –version" to ensure Python version 3.11 is installed and configured.
4. (Optional) If running the application in a virtual environment is desired, configure it and install the following libraries while it is activated.
5. In a command prompt window type: "pip install pandas"
6. In a command prompt window type: "pip install numpy:
7. In a command prompt window type: "pip install matplotlib"
8. In a command prompt window type: "pip install -U scikit-learn"
9. In a command prompt window type: "pip install PyQt5"

An example of how to use the application:

1. Extract the “C964_CS_Capstone_Project.zip” file that was submitted to any directory and make note of the path to it.
2. In a command prompt window, navigate to the directory where the “C964_CS_Capstone_Project.zip” file was extracted and go into that directory.
3. In a command prompt type “dir”, and verify that the “main.py”, “plot_data.py”, “process_and_train_data.py”, and “diabetes.csv” files are there, if any of those files are not present, go back to step 1.
4. If typing “python –version” in the command prompt window displays “Python 3.11.X”, continue to step 5a. If typing “python3 –version” in the command prompt window displays “Python 3.11.X, continue to step 5b.
    a. In a command prompt window, type: “python main.py” and ensure the applicationGUI is displayed.
    b. In a command prompt window, type: “python3 main.py” and ensure the applicationGUI is displayed.
5. Once the application is running:
    a. A label is shown informing the user if the “diabetes.csv” dataset has been found or not.
        i. If the “diabetes.csv” file is not found, the user will not be able to proceed and will be informed that the “diabetes.csv” file needs to be in the same directory as the application file.
        ii. If the “diabetes.csv” file is found, the user will be able to proceed.
    b. The “Select dataset to use:” drop down box can be used to either select the original, unmodified, dataset to proceed with, or a cleaned version of the original dataset where all “0” values are replaced with the average value of the column, for all columns except for the pregnancy column.
    c. The close button will close any open windows and quit the application.
6. The “Show Graphs” button will open a new window where the data can be graphed in either a bar graph, scatter plot, or a histogram graph.
    a. Use the “Select Data to Plot” drop down box to select which column of data to graph.
    b. The “Hide data with a value of 0” check box can be used to hide all “0” values for the scatter plot and histogram graph. Note: The “Hide data with a value of 0” check box will be hidden when “Pregnancies” is selected in the “Select Data to Plot” drop down box.
    c. Select a graph/plot to display the data by clicking on the corresponding button.
        i. The bar graph will display max value for the selected column, the average value of the dataset where all results are positive for diabetes, the average value of the dataset where all results are negative for diabetes, and the min value for the selected column.
        ii. The scatter plot will display all values in the selected column and will be green for results where the patient was negative for having diabetes and red for results where the patient was positive for having diabetes.

iii. The histogram graph will display all values in the selected column.

d. The “Close” button will close the Plot Data window.

7. The “Predict” button will open a new window where the user can enter data and get a prediction of whether diabetes is predicted or not.
    a. Either select a predefined set of patient data from the “Predefined Values” drop down box or enter custom values in the input spinboxes, then click on the “Make Prediction” button. The result will be displayed at the top of the window.
    b. Click the “Retrain Model” button to retrain the model.
    c. The “Training Model Accuracy” label to the right of the “Retrain Model” button will display the accuracy of the tested dataset.
    d. The “Close” button will close the Predict window.

Reference Page

1.

Author: Tableau

Date: Unknown

Title: What are The Top Machine Learning (ML) Methods?

Source: https://www.tableau.com/learn/articles/top-machine-learning-methods

2.

Author: Geeks for Geeks

Date: 27-Feb-2024

Title: Supervised Machine Learning

Source: https://www.geeksforgeeks.org/supervised-machine-learning/

3.

Author: Geeks for Geeks

Date: 17-May-2024

Title: Decision Tree

Source: https://www.geeksforgeeks.org/decision-tree/

4.

Author: UCI MACHINE LEARNING and Kaggle Team

Date: Unknown (Updated 8 years ago)

Title: Pima Indians Diabetes Database

Source: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

5.

Author: Huma Naz and Sachin Ahuja

Date: 19-June-2024

Title: Deep learning approach for diabetes prediction using PIMA Indian dataset

Source: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7270283/

6.

Author: Sruthy

Date: Updated March 9, 2024

Title: Validation Testing Ultimate Guide

Source: https://www.softwaretestinghelp.com/validation-testing/

7.

Author: Geeks for Geeks

Date: Last Updated - 22-Sep-2023

Title: Verification Methods in Software Verification

Source: https://www.geeksforgeeks.org/verification-methods-in-software-verification/

8.

Author: Ana Azevedo and M.F. Santos

Date: Unknown

Title: KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW

Source: https://recipp.ipp.pt/bitstream/10400.22/136/3/KDD-CRISP-SEMMA.pdf