Diabetes Predictor Tool Project

Mona Doane, # 011109368

Western Governors University

C964: Computer Science Capstone

January 20, 2024

Table of Contents

Table of Contents 2
Part A: Letter of Transmittal 4
Part B: Project Proposal Plan 6
Project Summary 6
Problem Summary 6
Customer Background 6
Vendor Background 6
Project Overview and Deliverables 7
Customer Outcomes 7
Data Summary 8
Implementation And Evaluation 9
CRISP-DM Methodology 10
Business Understanding: 10
Data Understanding 10
Data Preparation 10
Modeling: 11
Evaluation 11
Deployment 11
Agile Methodology 12
Outline of Proposed Implementation Process using Agile Principles 12
Timeline 14
Resources and Costs 15
Part C: Methodology and Application 16
Figure 1: 0-Diabetes_Predictor_Notebook.ipynb 17
Figure 2: 1-Methodology.ipynb 18
Figure 3: 2-Application.ipynb 18
Part D: Post-implementation Report 19
Solution Summary 19
Data Summary 20
Jupiter Notebook and Python Libraries (Step 1 of 1-Methodology.ipynb file) 20
Raw Data (Steps 2a and 2b of 1-Methodology.ipynb file) 20
Cleaned and Processed the Data (Steps 2c and 3 of 1-Methodology.ipynb file) 21

Example of the Data Cleaning Process 24
Visualized the data (Step 4 of 1-Methodology.ipynb file) 26
Bar Graph showing diabetes attribute versus an overweight attribute 26
Age Histogram 27
Correlation Matrix 28
Balanced and reduced the data (Step 5 of 1-Methodology.ipynb file) 29
Split the Cleaned Data into X (factors) and y (target) (Step 6 of 1-Methodology.ipynb file) 30
Created Training and Testing Data Sets (Step 6 of 1-Methodology.ipynb file) 30
Machine Learning 31
Selected Various Models for Evaluation (Step 7 of 1-Methodology.ipynb file) 31
Selected The Support Vector Machine Model as the basis for the Diabetes Predictor Tool 32
Finetuned the Support Vector Machine Model 32
Final Cleaned and Processed Data Set 34
Validation 35
Confusion Matrix 35
The Diabetes Predictor Tool 36
Diabetes Predictor Tool Demonstration 36
Preliminary User Guide 39
Running the Application for End-Users 39
Application Instructions 39
Alternate Instructions to Run Application on MyBinder.com 41
Running the Application and Methodology for Data Scientists 45
References 47

Part A: Letter of Transmittal

January 10, 2023

Jonathan Davies
Senior Vice President
First-Class Health Care Systems
2445 River Road
Colorado CO 80821

Dear Mr. Davies,

It was a pleasure meeting with you and your team earlier this week to discuss First-Class Health Care Systems' current needs and positioning in the healthcare marketplace. I appreciate your honesty in describing the challenges that exist in the diabetes healthcare space to provide effective medical care while experiencing staffing shortages and rising medical testing costs. As you know, my father died of complications from Type 2 Diabetes exacerbated by delayed diagnosis and medical testing, and it is my understanding that timely intervention could have saved his life. Knowing how critical decision-making tools are in the early diagnosis of Type 2 Diabetes, I am pleased to present to you the following proposal to develop a diabetes screening application to facilitate decision-making processes at your healthcare clinics.

The user-friendly self-serve diabetes screening application will consist of a 10-question survey for use by your front-line staff and your patients to determine the risk of Type 2 Diabetes and to facilitate timely diabetes evaluation and medical testing. The application will be driven by a powerful machine-learning model developed using publicly available data from the US Center for Disease Control and Prevention. The most current statistical algorithms and machine-learning techniques will be utilized to identify a model with an accuracy score of over 65%.

The Diabetes Predictor Tool Project will be conducted in a phased approach with five deliverables. The machine learning model, application, and user guide will be provided in the first three phases respectively using Cross Industry Standard Process for Data Mining (CRISP-DM) methodology. Once the application is developed, the implementation strategy will use Agile project management methodology in collaboration with First Class Health Care Systems to integrate the application into the existing patient portal and scheduling systems, thus ensuring that confidentiality and security of sensitive health information are maintained under HIPAA rules.

The estimated budget for this project is a one-time fee of $15,000. This cost provides 240 project hours using my expertise as a data scientist. While there is no charge for data acquisition, the fee includes costs associated with model training, software development, documentation, and integration into your existing technology. In addition, ABC Analytics offers 30 days of post-integration support for free. Ongoing support and maintenance are available at a preferred rate if desired.

My background in mathematics and computer science, my experience in data analytics and machine learning, and my commitment to the early diagnosis of Type 2 Diabetes means I am uniquely qualified to deliver this diabetes screening application to meet First-Class Health Care Systems' vision and objectives.

The Diabetes Screening Application is guaranteed to enhance your clinics' operational efficiency metrics while prioritizing Type 2 Diabetes patient scheduling and testing thereby positioning First-Class Health Care Systems for growth and success in the diabetes healthcare space.

I look forward to the opportunity to discuss this proposal further and explore how this data-driven solution can contribute to the achievement of First-Class Health Care Systems' objectives.

Thank you for your consideration.

Sincerely,

Mona Doane
Senior Data Analyst
ABC Analytics
mona_doane@ABCanalytics.com
888-555-5555 ext. 242

Part B: Project Proposal Plan

Project Summary

Problem Summary

Diabetes is a health condition that affects the basic function of how the human body turns food into energy. Type 2 Diabetes is the number one cause of kidney failure, blindness, and amputation in adults and the eighth-leading cause of death in the United States (Centers for Disease Control and Prevention, 2023). One of the critical factors in the treatment of diabetes requires timely diagnosis for intervention. Challenges faced by the healthcare industry include limited staffing and long wait times for expensive medical testing. These challenges result in the delay of diabetes diagnosis and treatment. The need to identify patients who have or may develop diabetes in the absence of medical evaluation provides an opportunity for a data-driven solution that can predict diabetes and prioritize the medical evaluation and treatment of such patients.

Customer Background

First-Class Health Care Systems is the premier provider of healthcare in the United States with over 700 healthcare clinics and lab centers. Appropriate levels of staffing in these healthcare facilities and timely scheduling of laboratory testing pose challenges to First-Class Health Care Systems that result in the delayed diagnosis and treatment of Type 2 Diabetes. To improve diabetes healthcare services with timely care coordination, First-Class Health Care Systems requires a diabetes screening tool to allow patients to self-screen for diabetes and prioritize the scheduling of lab and medical services in the absence of an existing diabetes diagnosis. The tool should be simple to use and allow both patients and medical staff to quickly identify the possibility of diabetes to prioritize in-office consultations and medical testing.

Vendor Background

ABC Analytics has extensive experience harnessing the power of healthcare data to develop tools that integrate seamlessly into existing technological infrastructure. ABC's leading data analyst Mona Doane has degrees in Mathematics from the University of Waterloo and in Computer Science from Western Governors University. She is committed to the early diagnosis of diabetes and has over 10 years of data analytics experience in the healthcare industry. She will use CRISP-DM methodology for data analysis and application development phases. For the application implementation phases, ABC Analytics recommends the use of the Agile methodology of project management which emphasizes the partnership and collaboration required to deploy the data product within First Class Health Care Systems proprietary web applications.

Project Overview and Deliverables

ABC Analytics will use machine learning on existing publicly available data to create an application that will use questions to predict the possibility of Diabetes. This is a classification problem that categorizes patients by likelihood of diabetes ("diabetes" or "no diabetes") and can be answered using supervised learning classification techniques. Once the publicly sourced data from the CDC is downloaded from Kaggle.com to ABC Analytics' local computing environment, it can be run through supervised machine learning algorithms to develop a model. Jupyter Notebook, a web-based open-source interactive computing platform that provides advanced Python Data Analysis Libraries such as NumPy, Pandas, Matplotlib, and Scikit-Learn (Sujan, 2018), will be used for the data analysis and model building and the Python module Ipywidgets will be used to deliver the preliminary interactive Diabetes Predictor Tool.

The Tool will seamlessly integrate into First-Class Health Care Systems' existing web applications, MyHealth® patient portal, and the HealthInTime® scheduling system, to prioritize at-risk diabetes patients leading to earlier intervention. ABC Analytics will facilitate the Tool's implementation within these existing web applications. As the Tool is developed with non-proprietary tools (open-source resources), security is not a concern for the development. Final implementation occurs in web applications maintained by First-Class Health Care Systems. Therefore, the security and privacy of the health data obtained will comply with HIPAA regulations using Veridify Security® technology which provides state-of-the-art cybersecurity solutions for health organizations.

Specifically, the deliverables for this Project are as follows:

1. Data analysis and model training methodology in the form of Jupyter Notebook file which includes visualizations used in data exploration, machine learning, model building, and evaluation.
2. Diabetes Predictor Tool proof-of-concept application in the form of a Jupyter Notebook file which includes user interaction functionality.
3. Post Implementation Report as a .pdf file containing User Instructions.
4. Integration of the Tool in HealthInTime® Scheduling System*.
5. Integration of the Tool in MyHealth® Patient Portal*.

*Integration of the Tool in First-Class Health Care Systems' proprietary web applications will be subject to the approval of the deliverables provided in 1, 2, and 3 and will not be included for evaluation in this report.

Customer Outcomes

First-Class Health Care Systems' patients will benefit from the implementation of this tool as they will be able to predict whether they may have diabetes or not. Front-line staff with limited medical training can use the Diabetes Predictor Tool as a decision-making tool to prioritize the scheduling of medical testing and office visits. This in turn will lead to earlier diagnosis and treatment of diabetes. More positive health outcomes in the diabetes healthcare space will position First-Class Health Care Systems as the preferred healthcare company in a market that includes 98 million pre-diabetes adults who remain undiagnosed (Centers for Disease Control and Prevention, 2022).

Data Summary

The raw data is available from the U.S. Center for Disease Control and Prevention (CDC) which uses the Behavioral Risk Factor Surveillance System (BRFSS) to annually conduct over 400,000 telephone surveys regarding health-related behaviors and chronic health conditions of U.S. residents (U.S. Centers for Medicare & Medicaid Services, 2023).

The BRFSS data is ideal for several reasons:

1. The data is stripped of any unique, personalized, or identifying healthcare data and thus meets the strict rules set forth by the Health Insurance Portability and Accountability Act (HIPAA) for the security, privacy, and confidentiality of health information.
2. The data is comprehensive in that it offers over 300 attributes for study to determine the appropriate risk factors associated with Type 2 Diabetes.
3. The data is unbiased in that stringent data-collecting techniques are used, such as asking the survey questions without modification (U.S. Centers for Medicare & Medicaid Services, 2023).
4. The data is publicly available and free, limiting costs associated with this project.
5. The data is annually updated by the CDC, which can be used for future updates to the Diabetes Predictor Tool.
6. The data includes the diabetes labeled data which makes it suitable for use with supervised machine learning.

For efficiency, a pre-processed version of the most recent data from the CDC will be obtained from Kaggle.com, a popular data science platform. The pre-processed version retains the labels used by the BRFSS and thus, the BRFSS Codebook for 2021 provides a reference for understanding each of the attributes. Although the data will require further processing, the Kaggle version of the CDC data will serve as an effective starting point for data analysis, reducing cleaning time.

INFORMATION	URL LOCATION
CDC Data	https://www.cdc.gov/brfss/annual_data/annual_2021.html
Kaggle Version of Data	https://www.kaggle.com/datasets/dariushbahrami/cdc-brfss-survey-2021
BRFSS Codebook 2021	https://www.cdc.gov/brfss/annual_data/2021/pdf/codebook21_llcp-v2-508.pdf

Implementation And Evaluation

The Diabetes Predictor Tool Project will use a phased approach and two distinct methodologies. Data scientists combine CRISP-DM methodology for data analysis with team-based Agile methodology for deployment to remarkable success (Hotz, 2023).

The project has five phases with deliverables due at the end of each phase.

- The first phase produces the machine learning model using the data obtained. The Cross Industry Standard Process for Data Mining (CRISP-DM) will be the basis for the data science portion of this project.
- The second phase involves the development of the application. As the application consists of a few lines of Python code, formal software development methodology is not required.
- The third phase calls for the documentation including the User Guide. Note that the Preliminary User Guide will be deployed once the Tool is implemented within the web applications in the final phases. Again, formal methodology is not required to develop the documentation due to the limited nature of the application.
- In the final two phases, First-Class Health Care Systems' IT division with the consulting expertise of ABC Analytics will implement the Diabetes Predictor Tool within the MyHealth® patient portal and the HealthInTime® scheduling system using Agile Methodology which is iterative, collaborative, and user-focused. The official User Guide for front-line staff and patients will be created during these phases and provided at the end of the Project.

Evaluation occurs at the end of the third phase and the fifth phase as follows:

- At the end of the third phase, First-Class Health Care Systems will review the methodology and the application to ensure that the model meets the business requirements. The model should predict the occurrence of diabetes with 65% accuracy. Precision score metrics must also exceed 65% to minimize false positives. The application should execute and return a response of "diabetes" or "no diabetes" based on the survey questions using machine learning.
- At the end of the fifth phase, First-Class Health Care Systems' will perform user acceptance testing to evaluate the functionality and performance of the tool within the web environment.

CRISP-DM Methodology

The Cross Industry Standard Process for Data Mining (CRISP-DM) is a data science process model consisting of six phases.

The six phases are business understanding, data understanding, data preparation, modeling, evaluation, and deployment (Hotz, 2023). The CRISP-DM model will be loosely followed as described below.

Business Understanding:

Business Understanding involves understanding the business objectives, assessing the current situation, determining data mining goals, and producing a project plan (Hotz, 2023). The **Project Summary** section of this report discussed previously effectively outlines First-Class Health Care Systems' business objectives and current needs and resources and puts forth ABC Analytics' data mining goals and project proposal.

Data Understanding

This step involves collecting the data, describing the data, exploring the data, and verifying the data quality (Hotz, 2023). The use of the BRFSS data means collection of data is limited to downloading the data from the Kaggle.com website. Using **head()**, **tail(),** and **shape ()** functions in Jupyter Notebook will reveal column names and the size of the data set. Functions such as **info()** and **describe()** will be used to determine memory usage and data distribution.

Data Preparation

This step includes cleaning the data by correcting, imputing, and removing erroneous and unusable data as well as formatting the data in preparation for model building.

Reviewing the BRFSS Codebook for 2021 provides explanations for the names of columns in the data corresponding to potential factors of interest. There are over 300 potential attributes and research will need to be conducted on this list of columns to a workable data set. The downloaded data will be revised to a data frame containing a selection of attributes of interest and the columns will be renamed using the **rename()** function for easier understanding.

Once the dataset is reduced to the above columns, any rows with unusable responses will be removed. The dataset contains NaN values which reflect missing data and incomplete information with responses such as "Don't know" and "Refused" which are not useful for machine learning. The data frame selection will be revised using the **dropna()** function to remove data with NaN values. The remaining data will be cleaned and streamlined into binary categories where possible.

Once the data has been thoroughly cleaned, visualization techniques using the matplotlib Python library will be used to describe relationships between diabetes and other variables such as BMI. Histograms will be used to study the date frame selection for biases or imbalances. Correlation matrixes will be used to identify impactful features for use in the model. Finally, the data frame selection will be modified to balance class data (diabetes versus no diabetes) to reduce class bias. This

set will be reduced in size to be determined based on trial and error to ensure appropriate response time for model training and evaluation.

Modeling:

This step involves selecting modeling techniques, generating test designs, building models, and assessing models (Hotz, 2023). The data provided in the BRFSS dataset contains both features (the variables other than diabetes) and the target known as labeled data (diabetes or no diabetes). Since the target is contained within the data set, supervised machine learning will be used to discover patterns and relationships between the features and the target in a process called fitting or training. Within supervised machine learning, classification algorithms that predict an event in the future having two discrete outcomes (diabetes or no diabetes) are called Binary Classification algorithms (Ali, 2022). Five common machine learning predictive models that are used for binary classification are the Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, and Bernoulli Naïve Bayes Classifiers (Fregoso-Aparicio, 2021). The data will be separated into training and testing sets and the training data will be used to train each model.

Evaluation

The three models will be evaluated based on accuracy scores and precision metrics obtained from running the test set of data. The selected model must be able to predict whether a person has diabetes based on the features with 65% accuracy and precision scores. While the model should be reasonably accurate (predicts diabetes when the person has diabetes and predicts no diabetes when the person does not have diabetes), the model should also be precise (Shung, 2018) as higher precision means fewer false positives (predicts diabetes when the person does not have diabetes). Maximizing precision is crucial for First-Class Health Care Systems as it strives to prioritize patients with diabetes with limited resources for diagnosis and treatment. The final data set will be finetuned using feature importance to remove less notable features. The final model will be built using the final data set. The confusion matrix will be reviewed to view actual versus predicted percentages of diabetes and no diabetes and the model will be evaluated.

Deployment

The Diabetes Predictor Tool will be built using Ipywidgets in a Jupyter notebook and use the machine learning model to make predictions. Deployment in this context means that First-Class Health Care Systems will be able to use and evaluate the Tool and determine steps using Agile methodology for integration of the Tool into their proprietary systems.

Agile Methodology

Agile is a project management and software development methodology that focuses on flexibility, collaboration, and responsiveness to change. Characteristics of Agile methodology consist of adaptive planning, cross-functional teams, transparent communication, customer collaboration, and time-boxed iterative incremental development using sprints focusing on delivery (Laoyan, 2022).

Agile is ideal for the integration of the Diabetes Predictor Tool into First-Class Health Care Systems as it provides the opportunity for feedback during the implementation process.

While a standard outline of the Agile Methodology used by ABC Analytics is provided here, it is subject to the approval of the project deliverables in phases 1-3. The Project Manager may modify it as needed. ABC Analytics' role in this portion of the project is to serve as the subject matter expert on the Diabetes Predictor Tool and the underlying performance metrics of the Tool within First-Class Health Care Systems web applications. For example, if the tool does not function quickly enough for the end-user, opportunities may exist to reduce data types to binary or utilize compression techniques to optimize the performance of the data model without sacrificing the accuracy of the model.

Outline of Proposed Implementation Process using Agile Principles

Objectives: Integrate the Diabetes Predictor Tool into the existing systems the MyHealth® patient portal and the HealthInTime® scheduling system.

Process: Times for each sprint are flexible due to the shortened timeframe for implementation.

1. **Sprint 1: Project Kickoff and Planning:** Conduct kickoff meeting with personnel listed under Key Roles. Confirm the project goals and establish success criteria for the implementation. Finalize the timeline and the objectives for sprints and the overall project.
2. **Sprint 2: Core Integration:** Develop an API for seamless communication and integration of the tool. Implement the basic functionality of the tool within the development environment of the web applications.
3. **Sprint 3: Testing and Refinement:** Conduct comprehensive testing of the integrated tool within the testing environment of the web applications. Refine the integrated tool based on user feedback. Optimize the performance of the application by reducing data storage requirements or model run time.
4. **Sprint 4: Deployment and Documentation:** Deploy the integrated application into the production environment in the web application. Prepare documentation and training materials for front-line staff. Conduct training sessions for end-users.

Customer Collaboration and Communication: Weekly sprint reviews and daily standup meetings will be conducted with the key team members and quality assurance personnel who will simulate patients and front-line staff to offer feedback on the performance of the Diabetes Predictor Tool within the web applications.

Key Roles: The cross-functional implementation team will consist of the following key personnel:

1. Scrum Master: Bryan Wilson, Project Manager
2. Product Owner: Jonathan Davies, SVP - Operations
3. Development Team:
    a. Web developer: Aaron Hayes
    b. API/Interface developer: Vikram Spartan
    c. Document coordinator: Amanda Hayes
    d. Data scientist: Mona Doane, ABC Analytics
4. Quality Assurance Team (QA) – To be determined

Success Criteria: Success Criteria for Phases 4 and 5 are as follows:

1. Successful integration of the Diabetes Predictor Tool within the MyHealth® patient portal and the HealthInTime® scheduling system
2. Positive feedback from First-Class Health Care Systems end-users (patients and front-line staff)
3. Improved time in identifying and scheduling potential diabetes patients based on risk factors.

Timeline

The project will be implemented in a phased approach. Based on my experience on similar projects, the anticipated timeline for completion of the project will be 6 weeks. It will take 3 weeks to complete data analysis and develop the data model. I expect another week to develop a proof-of-concept version of the application for your evaluation and produce the user guide and evaluation report.

Once the deliverables from phases 1, 2, and 3 are approved by First-Class Health Care Systems, schedules for Phases 4 and 5 will be finalized. Based on my discussions with your highly experienced IT staff, I estimate two weeks as sufficient for the integration of the Diabetes Predictor Tool within First-Class Health Care Systems' proprietary applications. I will provide my expertise to your IT team members to address any implementation concerns or facilitate any deployment issues during this time.

Phase	Milestone or deliverable	Duration (Hours)	Projected Start Date	Anticipated End Date
1	Data analysis and model training completion in the form of Jupyter Notebook Methodology which includes visualizations used in data exploration, machine learning, and model evaluation	160	December 15, 2023	January 12, 2024
2	Diabetes Predictor Tool proof-of-concept in the form of Jupyter Notebook Application and Evaluation Report delivery in the form of a .pdf file	8	January 12, 2024	January 13, 2024
3	Evaluation Report delivery in the form of a .pdf file containing User Instructions	8	January 14, 2024	January 15, 2024
4	Implementation and deployment of the Tool in HealthInTime® Scheduling System*	32	January 15, 2024	January 22, 2024
5	Implementation and Deployment of the Tool in MyHealth® Patient Portal*	40	January 22, 2024	January 31, 2024
	Total Project Duration	**240**	**December 15, 2023**	**January 31, 2024**

* Subject to First-Class Health Systems' final approval of deliverables from Phases 1 to 3.

Resources and Costs

The data is publicly available at no cost. The methodology and proof-of-concept application will be built on Jupyter Notebook and housed on GitHub which are both open-source and require no additional expenses. The proof-of-concept application can be run on Binder which is also open-source.

The project includes 240 hours billed at a rate of $62.50 per hour for my expertise in data analytics and software development as well as application rights upon completion.

This fee includes the following services:

- Sourcing the data
- Cleaning the data
- Visualization of the data
- Training and testing several machine learning models
- Evaluating and selection of the most precise model
- Finetuning the model
- Application Development
- Facilitation of Application Integration
- User Documentation

The total fee for this project is $15,000 with $10,000 due at the end of the third phase on January 19th and the remaining $5000 due at the end of the project.

As the machine learning model is derived from a public data set and the methodology used to create the model is publicly sourced, the model and the Diabetes Predictor Tool can be replicated without a license. However, the data generated from the Tool using First-Class Health Care System's proprietary software will remain the confidential information of First-Class Health Care Systems and its patients.

Part C: Methodology and Application

The Methodology and Application associated with this project are implemented in Jupyter Notebooks and available at the following site: https://github.com/mdoane7/capstone

The repository contains three Jupyter Notebooks.

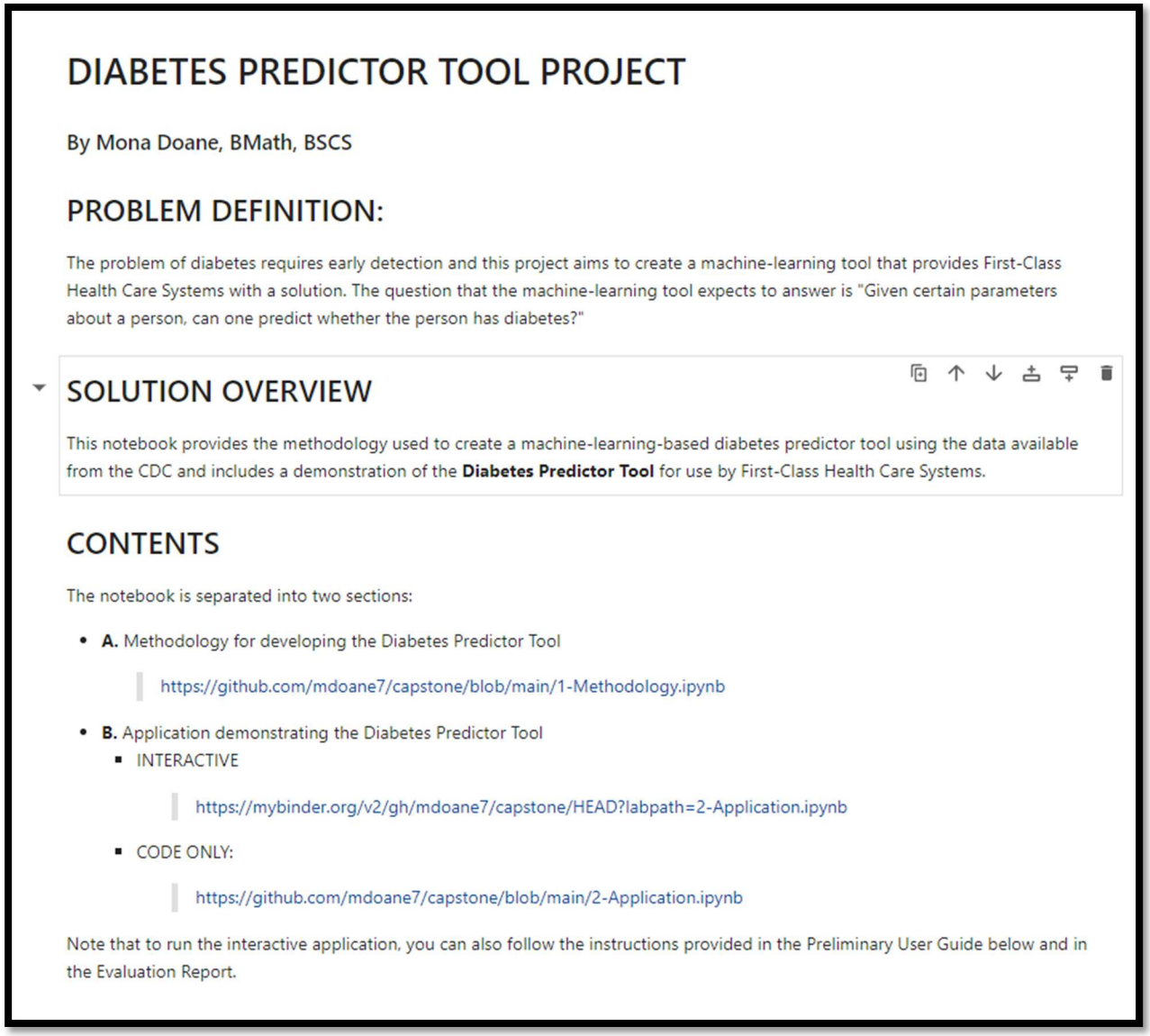
FILE NAME	DESCRIPTION OF FILE CONTENTS	LINKS TO VIEW OR INTERACT WITH FILES
0-Diabetes_Predictor_Notebook.ipynb	Contains a brief project overview with links to methodology and application notebooks.	GITHUB: (Viewable with links) https://github.com/mdoane7/capstone/blob/main/0-Diabetes%20Predictor%20Notebook.ipynb
1-Methodology.ipynb	Contains the methodology used to establish the machine learning model.	GITHUB: (Viewable with executed cells) https://github.com/mdoane7/capstone/blob/main/1-Methodology.ipynb
2-Application.ipynb	Contains the proposed tool for implementation utilizing the machine learning model.	MYBINDER: (Interactive) https://mybinder.org/v2/gh/mdoane7/capstone/HEAD?labpath=2-Application.ipynb GITHUB: (Viewable ONLY) https://github.com/mdoane7/capstone/blob/main/2-Application.ipynb
Final_Data.csv	Contains the cleaned data used to train the model.	GITHUB: https://github.com/mdoane7/capstone/blob/main/Final_Data.csv
Trained_Model.sav	Contains the model used to power the Diabetes Predictor Tool.	GITHUB: https://github.com/mdoane7/capstone/blob/main/Trained_Model.sav

The second file **1-Methodology.ipynb** is rendered with executed cells and can be viewed on GitHub.

The third file **2-Application.ipynb** should be run under the instructions provided in the User Guide section Application for End-Users which allows for user interaction if the link above does not execute on the user's system.

Sample images of the files are shown following for reference.

Figure 1: 0-Diabetes_Predictor_Notebook.ipynb



DIABETES PREDICTOR TOOL PROJECT

By Mona Doane, BMath, BSCS

PROBLEM DEFINITION:

The problem of diabetes requires early detection and this project aims to create a machine-learning tool that provides First-Class Health Care Systems with a solution. The question that the machine-learning tool expects to answer is "Given certain parameters about a person, can one predict whether the person has diabetes?"

SOLUTION OVERVIEW

This notebook provides the methodology used to create a machine-learning-based diabetes predictor tool using the data available from the CDC and includes a demonstration of the **Diabetes Predictor Tool** for use by First-Class Health Care Systems.

CONTENTS

The notebook is separated into two sections:

- **A.** Methodology for developing the Diabetes Predictor Tool

  > https://github.com/mdoane7/capstone/blob/main/1-Methodology.ipynb

- **B.** Application demonstrating the Diabetes Predictor Tool
  - INTERACTIVE

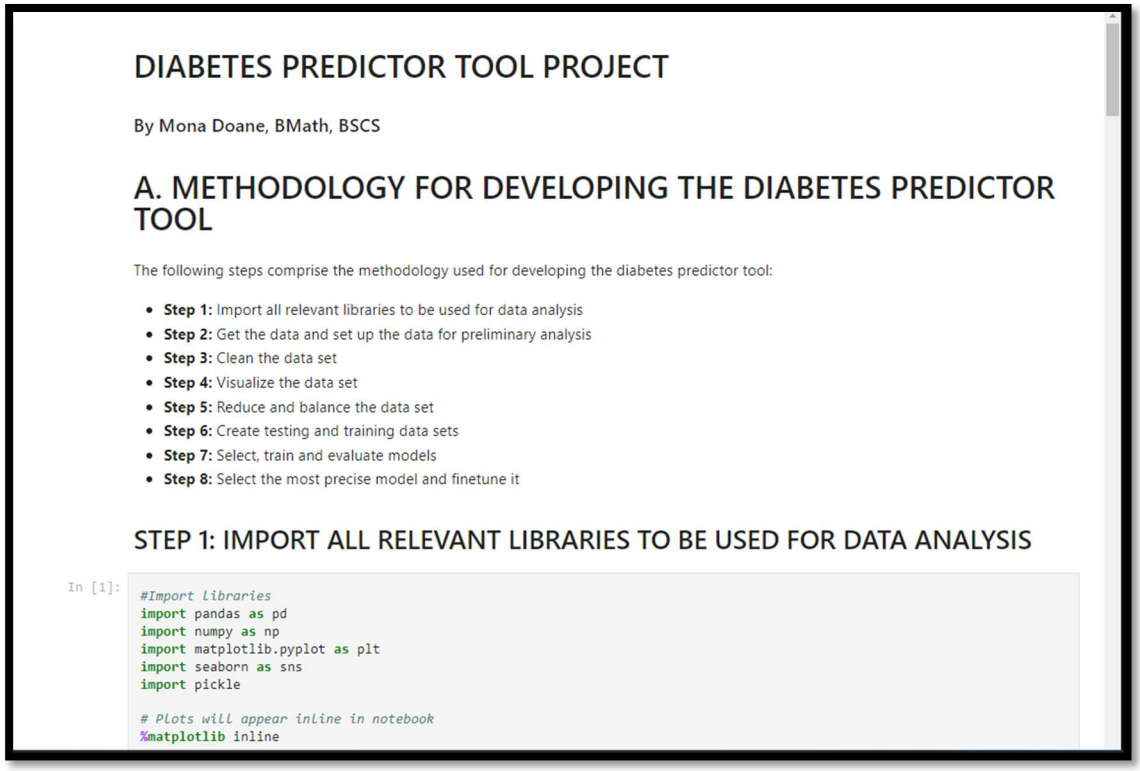
    > https://mybinder.org/v2/gh/mdoane7/capstone/HEAD?labpath=2-Application.ipynb

  - CODE ONLY:

    > https://github.com/mdoane7/capstone/blob/main/2-Application.ipynb

Note that to run the interactive application, you can also follow the instructions provided in the Preliminary User Guide below and in the Evaluation Report.

Figure 2: 1-Methodology.ipynb



DIABETES PREDICTOR TOOL PROJECT

By Mona Doane, BMath, BSCS

A. METHODOLOGY FOR DEVELOPING THE DIABETES PREDICTOR TOOL

The following steps comprise the methodology used for developing the diabetes predictor tool:

- **Step 1:** Import all relevant libraries to be used for data analysis
- **Step 2:** Get the data and set up the data for preliminary analysis
- **Step 3:** Clean the data set
- **Step 4:** Visualize the data set
- **Step 5:** Reduce and balance the data set
- **Step 6:** Create testing and training data sets
- **Step 7:** Select, train and evaluate models
- **Step 8:** Select the most precise model and finetune it

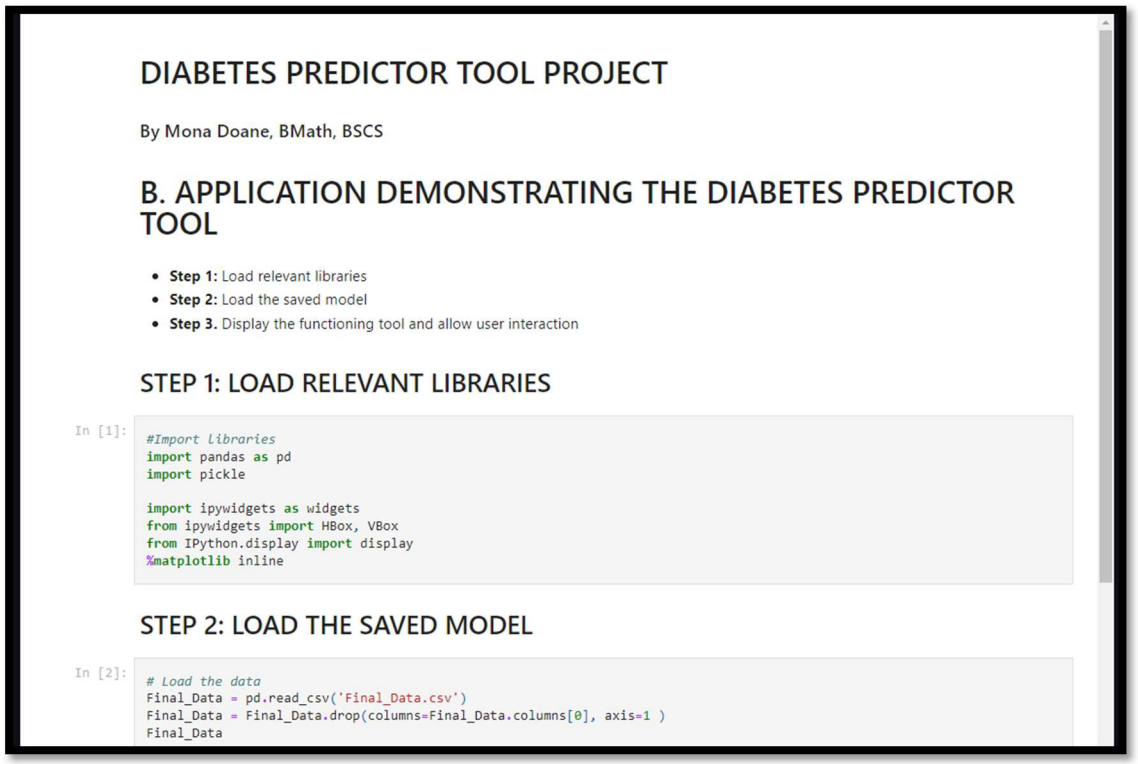
STEP 1: IMPORT ALL RELEVANT LIBRARIES TO BE USED FOR DATA ANALYSIS

In [1]:

```
#Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle

# Plots will appear inline in notebook
%matplotlib inline
```

Figure 3: 2-Application.ipynb



DIABETES PREDICTOR TOOL PROJECT

By Mona Doane, BMath, BSCS

B. APPLICATION DEMONSTRATING THE DIABETES PREDICTOR TOOL

- **Step 1:** Load relevant libraries
- **Step 2:** Load the saved model
- **Step 3.** Display the functioning tool and allow user interaction

STEP 1: LOAD RELEVANT LIBRARIES

In [1]:

```
#Import libraries
import pandas as pd
import pickle

import ipywidgets as widgets
from ipywidgets import HBox, VBox
from IPython.display import display
%matplotlib inline
```

STEP 2: LOAD THE SAVED MODEL

In [2]:

```
# Load the data
Final_Data = pd.read_csv('Final_Data.csv')
Final_Data = Final_Data.drop(columns=Final_Data.columns[0], axis=1 )
Final_Data
```

Part D: Post-implementation Report

Solution Summary

Diabetes is a disease that causes severe complications and mortality in the United States. Early diagnosis and treatment of diabetes lead to positive outcomes for diabetes patients. First-Class Health Care Systems, a premier healthcare provider, faced challenges in ensuring undiagnosed diabetes patients received timely healthcare services due to limited staffing and lab availability.

ABC Analytics produced a Diabetes Predictor Tool which used machine learning to predict whether a person had diabetes or not based on responses to a survey. The Tool predicted that patients had diabetes with over 70% accuracy and over 70% precision. The questions used in the diabetes predictor tool were based on the attributes used by the machine learning model. Drop-down boxes and sliders were used to minimize invalid data entries. Once the answers were completed and the user clicked the predict button, the Diabetes Predictor Tool predicted whether the patient had diabetes or not based on the machine learning model.

First-Class Health Care Systems implemented the diabetes screening tool in its proprietary systems improving the efficiency of diabetes diagnosis and treatment in their extensive healthcare network. The Tool used in the patient portal in a self-serve mode created a sense of urgency in its patients to go to the doctor and get medical testing leading to early diagnosis of diabetes. The Tool was also implemented within the scheduling system allowing medical personnel to prioritize testing and office visits for patients whom the Tool deemed had diabetes. Early diagnosis led to faster implementation of treatment options and diabetes management for these patients and more positive outcomes.

Data Summary

Jupiter Notebook and Python Libraries (Step 1 of 1-Methodology.ipynb file)

The following Python libraries were used to load, clean, process, and model the data in Jupyter Notebook. The libraries are listed here for reference.

[1]:
```
#Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle

# Plots will appear inline in notebook
%matplotlib inline

# Models from Scikit-Learn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import BernoulliNB
from sklearn.neighbors import KNeighborsClassifier

# Model Evaluations
from sklearn.inspection import permutation_importance
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.metrics import precision_score, recall_score, f1_score
```

Raw Data (Steps 2a and 2b of 1-Methodology.ipynb file)

While the original data was collected by the US Center for Disease Control and Prevention under the Behavioral Risk Factor and Surveillance System (BRFSS) in 2021, the data used for this project was downloaded from Kaggle.com as the data located in Kaggle is partially pre-processed and saved as a CSV file.

The data was renamed as Diabetes_Dataset_2021.csv and loaded into Jupyter Notebook using the Python library Pandas.

[2]:
```
# Load the dataset
CDC_Diabetes_DataSet = pd.read_csv('Diabetes_Dataset_2021.csv')
```

The data was inspected using the **shape()** command to determine how many rows and columns were included.

```
[3]: # View the number of rows and columns in dataset
     CDC_Diabetes_DataSet.shape
[3]: (438693, 303)
```

COMMENTS

> There are 438,693 rows and 303 columns. I will need to reduce the number of rows and identify key features (or columns) to train the machine model without losing accuracy.

The **head()** command was used to view the first five rows of data.

```
[4]: # View the dataset to gain a basic understanding of the data
     CDC_Diabetes_DataSet.head()
```

[4]:

	_STATE	FMONTH	IDATE	IMONTH	IDAY	IYEAR	DISPCODE	SEQNO	_PSU	CTELENM1	...	_FRTRES1	_VEGRES1	_FRUTSU1	_VEGESU1	_FRTLT1A	_V
0	1	1	1192021	1	19	2021	1100	2021000001	2021000001	1.0	...	1	1	100.0	214.0	1	
1	1	1	1212021	1	21	2021	1100	2021000002	2021000002	1.0	...	1	1	100.0	128.0	1	
2	1	1	1212021	1	21	2021	1100	2021000003	2021000003	1.0	...	1	1	100.0	71.0	1	
3	1	1	1172021	1	17	2021	1100	2021000004	2021000004	1.0	...	1	1	114.0	165.0	1	
4	1	1	1152021	1	15	2021	1100	2021000005	2021000005	1.0	...	1	1	100.0	258.0	1	

5 rows × 303 columns

The **info()** command was used to determine memory usage.

```
[5]: # Attempt to get some information on the data
     CDC_Diabetes_DataSet.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 438693 entries, 0 to 438692
Columns: 303 entries, _STATE to _VEGETE1
dtypes: float64(245), int64(58)
memory usage: 1014.1 MB
```

Cleaned and Processed the Data (Steps 2c and 3 of 1-Methodology.ipynb file)

The data set contained 303 attributes or potential features. As such a large number of columns presented challenges to resource capabilities, the strategy to wrangle the data into a workable dataset involved narrowing down the attributes. Specifically, the focus was to reduce the number of columns to less than 20 attributes known to correlate with Diabetes.

Research was conducted using government websites and peer-reviewed journals to determine appropriate attributes of interest. Risk factors commonly associated with diabetes, particularly Type 2 Diabetes and Prediabetes include age, sex, race, weight (obesity and Body-Mass Index indicators), and low level of physical activity (Centers for Disease Control and Prevention, 2022). The presence of cardiac and stroke events, high blood pressure, and high cholesterol as well as dietary factors such as fruits, vegetables, and alcohol showed some correlation with the occurrence of diabetes (Hussein, 2022).

Using this research, the list of attributes was narrowed to the following fourteen columns.

DIABETES AND ITS POSSIBLE RISK FACTORS

1. Diabetes = DIABETE4
2. High Blood Pressure = _RFHYPE6
3. High Cholesterol = TOLDHI3
4. BMI = _BMI5
5. Smoke = SMOKE100
6. Stroke = CVDSTRK3
7. Heart Issue = _MICHD
8. Physical Activity = _TOTINDA
9. Fruit = _FRTLT1A
10. Vegetable = _VEGLT1A
11. Alcohol = DRNKANY5
12. Sex = _SEX
13. Age = _AGEG5YR
14. Race = _PRACE1

The above columns were retained using the following command.

```
[7]: # Narrow 303 columns to 16 columns that are the most interesting for study
CDC_Diabetes_DataFrame_Selection = CDC_Diabetes_DataSet[['DIABETE4', '_RFHYPE6', 'TOLDHI3', '_BMI5', 'SMOKE100', 'CVDSTRK3',
                                                         '_MICHD', '_TOTINDA', '_FRTLT1A', '_VEGLT1A', 'DRNKANY5', '_SEX',
                                                         '_AGEG5YR',  '_PRACE1']]
```

The downloaded data was renamed with the following columns using the **rename()** function.

```
[11]: #Rename the columns to make them more readable
CDC_Diabetes_DataFrame_Selection = CDC_Diabetes_DataFrame_Selection.rename(columns = {
    'DIABETE4':'Diabetes', '_RFHYPE6':'HighBloodPressure', 'TOLDHI3':'HighCholesterol',
    '_BMI5':'BMI', '_BMI5CAT':'Overweight', 'SMOKE100':'Smoker', 'CVDSTRK3':'Stroke', '_MICHD':'HeartIssues',
    '_TOTINDA':'PhysicalActivity','_FRTLT1A':'Fruits', '_VEGLT1A':"Vegetables", 'DRNKANY5':'Alcohol',
    '_SEX':'Sex', '_AGEG5YR':'Age', '_PRACE1':'Race'})
```

Once the columns were renamed, the **head()** command was used to view the data.

	Diabetes	HighBloodPressure	HighCholesterol	BMI	Smoker	Stroke	HeartIssues	PhysicalActivity	Fruits	Vegetables	Alcohol	Sex	Age	Race
0	3.0	1	1.0	1454.0	1.0	2.0	2.0	2	1	1	2	2	11	1.0
1	1.0	2	1.0	NaN	2.0	2.0	1.0	1	1	1	2	2	10	2.0
2	1.0	2	2.0	2829.0	2.0	2.0	1.0	2	1	2	2	2	11	2.0
3	1.0	2	1.0	3347.0	2.0	2.0	2.0	1	1	1	1	2	9	1.0
4	1.0	1	1.0	2873.0	2.0	1.0	1.0	1	1	1	2	1	12	1.0

NaN values cannot be used to train the model.

The data was inspected for these types of missing values using the **isna().sum()** command.

```
[13]: # Review to see if there are any missing values (i.e. Na)
      CDC_Diabetes_DataFrame_Selection.isna().sum()
```

```
[13]: Diabetes                3
      HighBloodPressure       0
      HighCholesterol     60836
      BMI                 46852
      Smoker              21232
      Stroke                  2
      HeartIssues          4635
      PhysicalActivity        0
      Fruits                  0
      Vegetables              0
      Alcohol                 0
      Sex                     0
      Age                     0
      Race                    4
      dtype: int64
```

The NaN values were dropped from the data using the **dropna()** command and the data was reviewed using **isna().sum()** command to verify it worked.

```
[14]: # Drop all missing data points
      CDC_Diabetes_DataFrame_Selection = CDC_Diabetes_DataFrame_Selection.dropna()
```

```
[15]: # Review to ensure rows with missing values (i.e. NaN) are dropped
      CDC_Diabetes_DataFrame_Selection.isna().sum()
```

```
[15]: Diabetes            0
      HighBloodPressure   0
      HighCholesterol     0
      BMI                 0
      Smoker              0
      Stroke              0
      HeartIssues         0
      PhysicalActivity    0
      Fruits              0
      Vegetables          0
      Alcohol             0
      Sex                 0
      Age                 0
      Race                0
      dtype: int64
```

Next, the BRFSS codebook was reviewed to get an understanding of the data. Each attribute was inspected in the BRFSS Codebook 2021 to ensure the data contained relevant information for the machine learning algorithms.

Attributes were found to contain values such as “Don’t know” and “Refused” which are not useful for machine learning. Rows with these values were removed.

The remaining data was modified to binary values where multiple categories existed. Body-mass index (BMI) values were processed to integer values between 0 and 100. Age categories were preserved in the thirteen categories reflecting 5-year intervals. The Race attribute was modified to six values: 5 common single-race denominations and the 6th for all other types of responses including "no response", "other", "multiracial" and "Don't know".

Step 3c in the Jupyter Notebook file **1-Methodology.ipynb** reflected the complete cleaning process for all fourteen factors. An example of the cleaning process is shown here demonstrating the algorithm used.

Example of the Data Cleaning Process

Table 1: Source: BRFSS Codebook (U.S. Centers for Medicare & Medicaid Services, 2023)

Label: (Ever told) you had diabetes
Section Name: Chronic Health Conditions
Core Section Number: 7
Question Number: 11
Column: 129
Type of Variable: Num
SAS Variable Name: DIABETE4
Question Prologue:
Question: (Ever told) (you had) diabetes? (If 'Yes' and respondent is female, ask 'Was this only when you were pregnant?'. If Respondent says pre-diabetes or borderline diabetes, use response code 4.)

Value	Value Label	Frequency	Percentage	Weighted Percentage
1	Yes	57,616	13.13	11.36
2	Yes, but female told only during pregnancy—Go to Section 08.01 HAVARTH5	3,808	0.87	0.99
3	No—Go to Section 08.01 HAVARTH5	366,342	83.51	85.04
4	No, pre-diabetes or borderline diabetes—Go to Section 08.01 HAVARTH5	9,942	2.27	2.40
7	Don't know/Not Sure—Go to Section 08.01 HAVARTH5	613	0.14	0.15
9	Refused—Go to Section 08.01 HAVARTH5	369	0.08	0.07
BLANK	Not asked or Missing	3	.	.

The information in the BRFSS Codebook reflected that the diabetes column included values of 1, 2, 3, 4, 7, and 9. As the Diabetes Predictor Tool only needed to be able to predict diabetes, the data needed to be streamlined.

The data was modified to a binary classification as follows:

- 0 (no diabetes) included values of 3 (No) and 2 (gestational diabetes),
- 1 (diabetes) included values of 1 (Yes), 4 (prediabetes),
- Values of 7 (Don't know) and 9 (Refused) were deleted, and

Note: Values of BLANK were deleted in earlier steps. Then the diabetes column was verified to only contain 0 and 1.

The code below reflects the cleaning process and the resulting output.

1. Variable: Diabetes

```
[18]: # This is what the survey will predict using machine learning
      # DIABETE4 = Diabetes = Have you ever been told you have diabetes?
      # Respondent values: 1 = Yes; 2 = Yes when pregnant; 3 = No; 4 = No but has pre-diabetes; 7 = Don't know; 9 = Refused
      # Revised dataset as follows:
          # Modified to binary classification:
              # 0 = No Diabetes (includes 2 = Yes when pregnant; and 3 = No)
              # 1 = Diabetes (includes 1 = Yes; and 4 = No but has pre-diabetes)
          # Removed: 7 = Don't know; and 9 = Refused
      CDC_Diabetes_DataFrame_Selection['Diabetes'] = CDC_Diabetes_DataFrame_Selection['Diabetes'].replace({1:1, 2:0, 3:0, 4:1})
      CDC_Diabetes_DataFrame_Selection = CDC_Diabetes_DataFrame_Selection[CDC_Diabetes_DataFrame_Selection.Diabetes != 7]
      CDC_Diabetes_DataFrame_Selection = CDC_Diabetes_DataFrame_Selection[CDC_Diabetes_DataFrame_Selection.Diabetes != 9]

      #Display all unique values in Diabetes Column (should be only 0s and 1s)
      CDC_Diabetes_DataFrame_Selection.Diabetes.unique()

[18]: array([0., 1.])
```

Once all the data was cleaned in this manner, different visualization techniques were used to study the data.

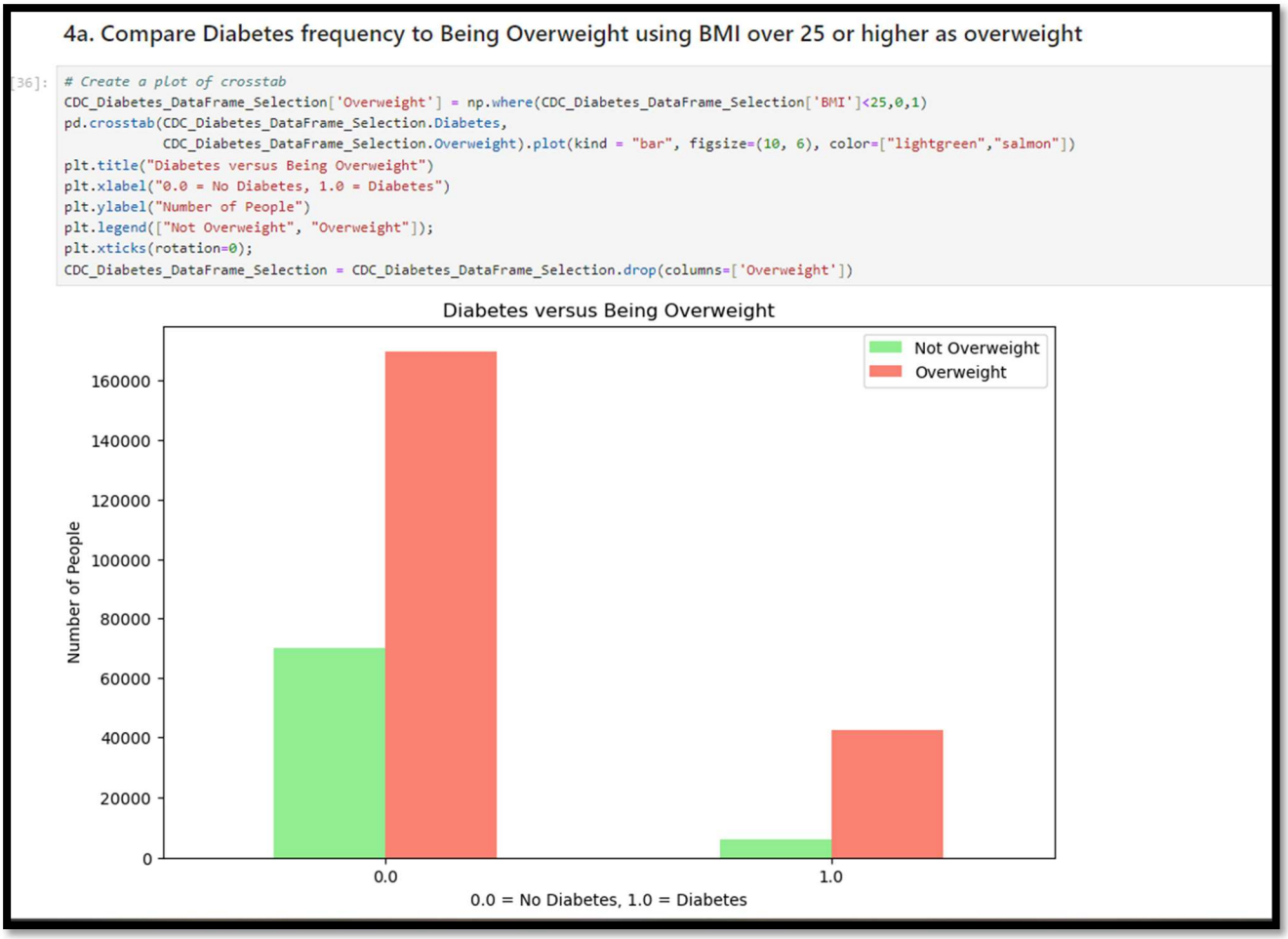
Visualized the data (Step 4 of 1-Methodology.ipynb file)

Bar Graph showing diabetes attribute versus an overweight attribute

Body-mass index (BMI) of 25 or over is defined by the CDC as overweight (U.S. Centers for Medicare & Medicaid Services, 2023). An overweight column was created and classified as 0 for not overweight defined by BMI under 25 and 1 for overweight defined by BMI 25 or over. It was obvious that a correlation existed between BMI and Diabetes. More people with diabetes were "overweight" than "not overweight." It was noted that the data was not balanced with regard to the diabetes attribute as there were more people without diabetes than people with diabetes. Therefore, further steps to clean the data were taken after visualization of the data.

4a. Compare Diabetes frequency to Being Overweight using BMI over 25 or higher as overweight

```
[36]: # Create a plot of crosstab
CDC_Diabetes_DataFrame_Selection['Overweight'] = np.where(CDC_Diabetes_DataFrame_Selection['BMI']<25,0,1)
pd.crosstab(CDC_Diabetes_DataFrame_Selection.Diabetes,
            CDC_Diabetes_DataFrame_Selection.Overweight).plot(kind = "bar", figsize=(10, 6), color=["lightgreen","salmon"])
plt.title("Diabetes versus Being Overweight")
plt.xlabel("0.0 = No Diabetes, 1.0 = Diabetes")
plt.ylabel("Number of People")
plt.legend(["Not Overweight", "Overweight"]);
plt.xticks(rotation=0);
CDC_Diabetes_DataFrame_Selection = CDC_Diabetes_DataFrame_Selection.drop(columns=['Overweight'])
```

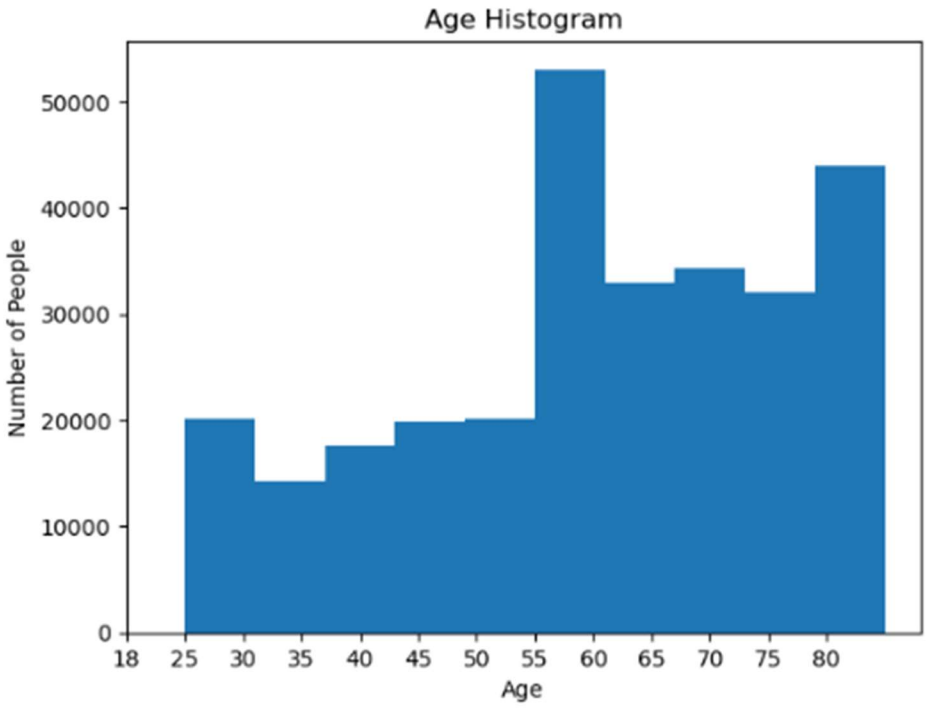


Age Histogram

The age histogram for the entire dataset showed that the data contained more people who are 55 and up than it did people less than age 55. As diabetes has a known correlation with age, this bias was acceptable for the dataset without the need to correct it or balance it.

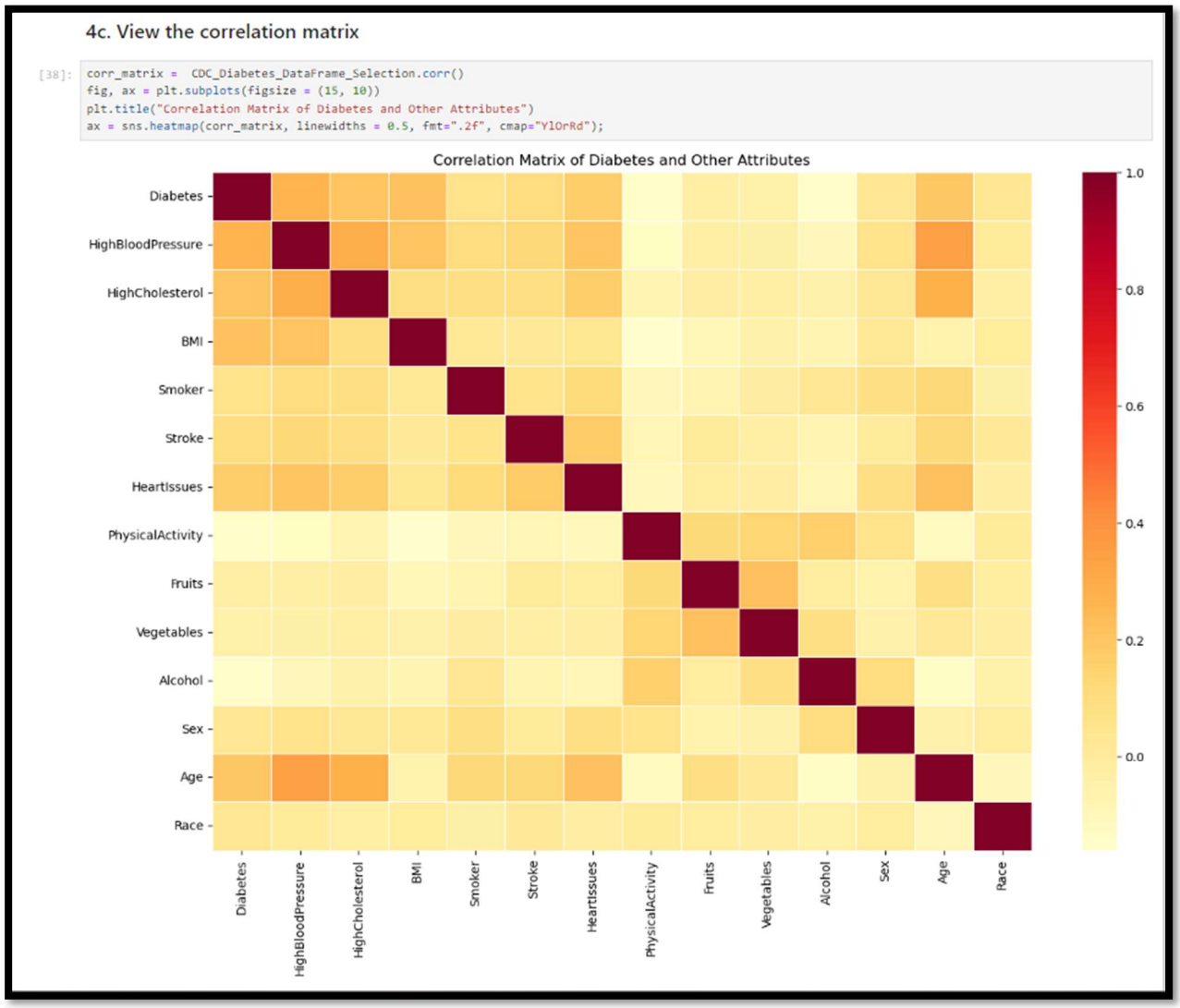
4b. View histograms of age of the complete dataset

```
[37]: CDC_Diabetes_DataFrame_Selection.Age.plot.hist()
labels= ['18', '25', '30', '35', '40', '45', '50', '55', '60', '65', '70', '75', '80']
plt.title('Age Histogram')
plt.xlabel('Age')
plt.ylabel('Number of People')
plt.xticks(np.arange(13), labels);
```



Correlation Matrix

The correlation matrix is an elegant way to visualize the correlation between diabetes and other attributes. The correlation matrix was depicted with graduated coloring from pale yellow to dark red to show the correlation between the different attributes. Diabetes showed a stronger correlation with age, high blood pressure, high cholesterol, heart issues, and BMI.



Balanced and reduced the data (Step 5 of 1-Methodology.ipynb file)

After visualization, the data was reviewed to see if it was appropriate for machine learning algorithms. Using **shape(),** it was determined the dataset consisted of 288,270 data points. More data points do not translate to a more accurate model and use excessive time and resources to train the model. Note that trial runs of sample sizes of 20,000, 40,000, and 80,000 were conducted but only improved the model by a maximum of 1.9%. To maintain resource efficiency without significant loss of accuracy or precision, a smaller sample size of 12,000 samples was randomly selected from the larger data set.

```
[39]: # Determine the size of data
      CDC_Diabetes_DataFrame_Selection.shape

[39]: (288270, 14)
```

The model must maintain impartiality in predicting both diabetes and non-diabetes outcomes. However, as noted in the visualization of Overweight versus Diabetes Bar Graph, the imbalance in the dataset with a higher representation of people without diabetes compared to those with diabetes, posed a challenge to the model's accuracy in predicting diabetes. Using **groupby().size()**, it was evident that there was a significant imbalance in classes of diabetes versus no diabetes.

```
1]: # Find the number of people with Diabetes and people without Diabetes in the data set.
    CDC_Diabetes_DataFrame_Selection.groupby(['Diabetes']).size()

1]: Diabetes
    0.0    239889
    1.0     48381
    dtype: int64
```

The following code was used to balance and reduce the dataset by selecting a random subset of the data.

5c. Balance the data set and reducing the dataset

```
[42]: # Separate the sets into 0s for No Diabetes and 1s for Pre-diabetes and Diabetes
      # Get the data for those who have Diabetes
      Have_Diabetes = CDC_Diabetes_DataFrame_Selection['Diabetes'] == 1
      Have_Diabetes_Data = CDC_Diabetes_DataFrame_Selection[Have_Diabetes]

      # Get the data for those who do not have Diabetes
      Does_Not_Have_Diabetes = CDC_Diabetes_DataFrame_Selection['Diabetes'] == 0
      Does_Not_Have_Diabetes_Data = CDC_Diabetes_DataFrame_Selection[Does_Not_Have_Diabetes]

[43]: # Select random data limited to the maximum number specified from each set
      Random_Does_Not_Have_Diabetes_Subset = Does_Not_Have_Diabetes_Data.take(np.random.permutation(len(Does_Not_Have_Diabetes_Data))[:MaxLength])
      Random_Have_Diabetes_Subset = Have_Diabetes_Data.take(np.random.permutation(len(Have_Diabetes_Data))[:MaxLength])

      # Join both sets with people with Diabetes and people without Diabetes.
      CDC_Diabetes_DataFrame_Balanced  = Random_Does_Not_Have_Diabetes_Subset._append(Random_Have_Diabetes_Subset, ignore_index = True)

[44]: # Check that the above resulted in a balanced set of data. The number of diabetes data points should equal the number of non-diabetes data points.
      CDC_Diabetes_DataFrame_Balanced.groupby(['Diabetes']).size()

[44]: Diabetes
      0.0    6000
      1.0    6000
      dtype: int64
```

As shown in line 44, the number of individuals with diabetes versus individuals without diabetes was equalized and the total dataset size was 12,000.

Split the Cleaned Data into X (factors) and y (target) (Step 6 of 1-Methodology.ipynb file)

The cleaned data was then split into two sets: one set reflecting the target variable y or "diabetes" and the other set reflecting all other variables X using the following code.

```
[46]: # Split data into X which reflects the matrix of independent variables and y is the set of target values to be predicted
      # X includes the data from columns other than diabetes
      X = CDC_Diabetes_DataFrame_Balanced.drop("Diabetes", axis = 1)

      # y includes the data from the diabetes column
      y = CDC_Diabetes_DataFrame_Balanced["Diabetes"]
```

Y contained the target variable "diabetes" shown below which is defined as dependent:

```
[48]: # Visualize y (Should just be the Diabetes column indicating whether someone has Diabetes(1) or not(0))
      y
```

```
[48]: 0        0.0
      1        0.0
      2        0.0
      3        0.0
      4        0.0
              ...
      11995    1.0
      11996    1.0
      11997    1.0
      11998    1.0
      11999    1.0
      Name: Diabetes, Length: 12000, dtype: float64
```

X contained all other variables which are defined as independent:

```
[47]: #Visualize X (Should reflect all columns except Diabetes column)
      X
```

[47]:

	HighBloodPressure	HighCholesterol	BMI	Smoker	Stroke	HeartIssues	PhysicalActivity	Fruits	Vegetables	Alcohol	Sex	Age	Race
0	0	1.0	21.0	0.0	0.0	0.0	1	0	1	0	0	13	1.0
1	1	0.0	35.0	0.0	0.0	0.0	0	1	1	0	0	12	1.0
2	1	0.0	28.0	1.0	0.0	0.0	1	0	1	1	0	12	1.0
3	1	1.0	39.0	1.0	0.0	0.0	1	1	1	0	0	8	1.0
4	0	0.0	29.0	1.0	0.0	0.0	1	1	0	1	1	12	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11995	1	1.0	37.0	0.0	0.0	0.0	0	1	1	0	0	11	1.0
11996	1	0.0	35.0	1.0	0.0	0.0	0	0	0	1	1	10	1.0
11997	0	0.0	30.0	1.0	0.0	0.0	0	0	1	1	1	8	1.0
11998	0	0.0	20.0	0.0	0.0	0.0	1	1	1	0	0	11	4.0
11999	1	1.0	31.0	0.0	0.0	1.0	1	1	1	0	0	11	1.0

12000 rows × 13 columns

Created Training and Testing Data Sets (Step 6 of 1-Methodology.ipynb file)

Once split, X and y were then divided randomly into training and testing sets. 80% of the data was used for training and 20% of the data was used for testing.

```
[49]: # Split data into training and testing sets; I will use a 80/20 split where 80% is used for training and 20% is used to testing
      X_train, X_test, y_train, y_test = train_test_split(X, y, stratify = y, test_size = 0.2, random_state=seed)
```

Machine Learning

Selected Various Models for Evaluation (Step 7 of 1-Methodology.ipynb file)

To develop the model underlying the Diabetes Predictor Tool, the five supervised machine learning methods chosen for evaluation were the Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, and Bernoulli Naïve Bayes Classifiers. The data is primarily binary, and these models lend themselves to the prediction of a class value given binary values (Fregoso-Aparicio, 2021).

```
[52]: # Create a list of models to test
      Models = []
      Models.append(('DT', DecisionTreeClassifier()))
      Models.append(('RF', RandomForestClassifier()))
      Models.append(('SVM', SVC(kernel='rbf')))
      Models.append(('KNN', KNeighborsClassifier()))
      Models.append(('BNB', BernoulliNB()))
```

Using the Scikit-Learn library, 80% of the data was used to train the models. 20% of the data was used to evaluate the models to determine the precision and accuracy metrics of the model in predicting diabetes.

The following excerpt from Jupiter Notebook reflects the code and resulting precision and accuracy scores.

```
[53]: # Train and Test the selected models
      Names = []
      Precision_Scores = []
      Accuracy_Scores = []
      for name, model in Models:
          model.fit(X_train, y_train)
          Accuracy_Scores.append(model.score(X_test, y_test)*100)
          Precision_Scores.append((precision_score(y_test, model.predict(X_test)))*100)
          Names.append(name)
      ModelsWithScores = pd.DataFrame({'Model': Names, 'Accuracy_Score': Accuracy_Scores, 'Precision_Score': Precision_Scores})
      print(ModelsWithScores)
```

```
  Model  Accuracy_Score  Precision_Score
0    DT       62.125000        62.707424
1    RF       67.458333        67.158067
2   SVM       71.791667        70.224285
3   KNN       68.208333        69.285084
4   BNB       69.375000        69.554247
```

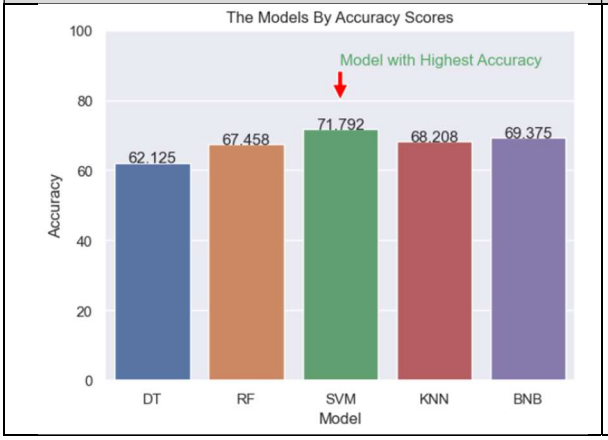
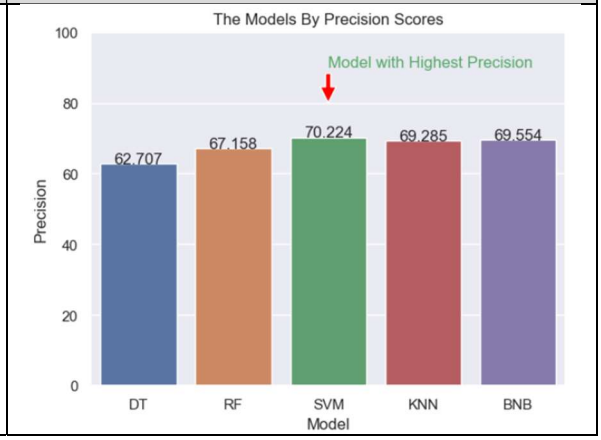
COMMENT

> The above accuracy scores and precision scores are on a scale of 1 to 100, with the higher number being better.

FOR REFERENCE:

1. DT = Decision Tree Classifier
2. RF = Random Forest Classifier
3. SVM = Support Vector Machine Classifier
4. KNN = K Nearest Neighbor Classifier
5. BNB = Bernoulli Naive Bayes Classifier

Displaying the scores in a graph demonstrated that the Support Vector Machine Classifier algorithm produced the model with the highest accuracy and precision.

ACCURACY SCORES	PRECISION SCORES
	

Selected The Support Vector Machine Model as the basis for the Diabetes Predictor Tool

The support vector machine (SVM) algorithm is a supervised learning method used for solving classification problems and regression tasks (IBM Corporation, 2021). In particular, SVM is commonly used for binary classification tasks in the biomedical industry, where the goal is to predict whether an individual has a particular disease or not (Yu, 2009). SVM is ideal for analyzing data with very large numbers (IBM Corporation, 2021) as well as when the sample size is small, and a large number of variables are involved (Yu, 2009). The SVM algorithm finds a hyperplane that maximally separates data into different classes (diabetes and not diabetes) while minimizing classification errors (Yu, 2009). The SVM model offered an accuracy score of 71.7917 and a precision score of 70.2243. The model was then evaluated to determine if it could be further improved.

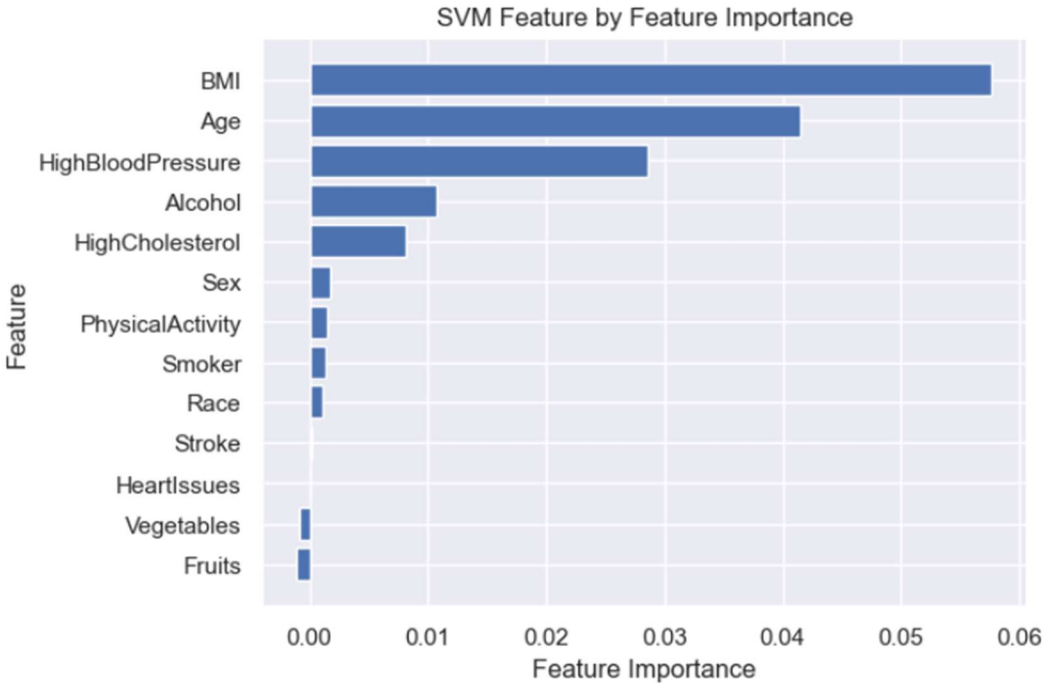
Finetuned the Support Vector Machine Model

The features of the SVM model were reviewed and evaluated using **permuation_importance** to determine feature importance to the accuracy of the model.

```
[60]: # Get all feature names
      Feature_Names = X.columns.values.tolist()
      Feature_Names

[60]: ['HighBloodPressure',
       'HighCholesterol',
       'BMI',
       'Smoker',
       'Stroke',
       'HeartIssues',
       'PhysicalActivity',
       'Fruits',
       'Vegetables',
       'Alcohol',
       'Sex',
       'Age',
       'Race']
```

```
[61]: # Display a bar graph of features by importance
Features = np.array(Feature_Names)
Feature_Importance = permutation_importance(Selected_Model_Before_Finetuning, X_test, y_test)
Sorted_Features_Index = Feature_Importance.importances_mean.argsort()
plt.barh(Features[Sorted_Features_Index], Feature_Importance.importances_mean[Sorted_Features_Index])
plt.xlabel("Feature Importance")
plt.ylabel("Feature")
plt.title("SVM Feature by Feature Importance")
plt.show()
```



The SVM model was finetuned to reduce any features that did not contribute positively to the model, namely vegetables and fruits, using the following command:

```
[62]: # Finalize the Data by dropping columns with lower impacts to reduce questions to ask on survey
Final_Data = CDC_Diabetes_DataFrame_Balanced.drop(columns=['Fruits','Vegetables'])
```

Final Cleaned and Processed Data Set

[2]:

	Diabetes	HighBloodPressure	HighCholesterol	BMI	Smoker	Stroke	HeartIssues	PhysicalActivity	Alcohol	Sex	Age	Race
0	0.0	0	1.0	21.0	0.0	0.0	0.0	1	0	0	13	1.0
1	0.0	1	0.0	35.0	0.0	0.0	0.0	0	0	0	12	1.0
2	0.0	1	0.0	28.0	1.0	0.0	0.0	1	1	0	12	1.0
3	0.0	1	1.0	39.0	1.0	0.0	0.0	1	0	0	8	1.0
4	0.0	0	0.0	29.0	1.0	0.0	0.0	1	1	1	12	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...
11995	1.0	1	1.0	37.0	0.0	0.0	0.0	0	0	0	11	1.0
11996	1.0	1	0.0	35.0	1.0	0.0	0.0	0	1	1	10	1.0
11997	1.0	0	0.0	30.0	1.0	0.0	0.0	0	1	1	8	1.0
11998	1.0	0	0.0	20.0	0.0	0.0	0.0	1	0	0	11	4.0
11999	1.0	1	1.0	31.0	0.0	0.0	1.0	1	0	0	11	1.0

12000 rows × 12 columns

This data was saved in a CSV file using the following command:

[74]:
```
# Save the data for use ine application
Final_Data.to_csv('Final_Data.csv')
```

The final data was used to retrain the SVM model.

[65]:
```
# Finalize the model
Final_Model = SVC(kernel='rbf')
```

[66]:
```
# Split data into X which reflects the matrix of independent variables and y is the set of target values to be predicted
# X includes the data from columns other than diabetes
X = Final_Data.drop("Diabetes", axis = 1)

# y includes the data from the diabetes column
y = Final_Data["Diabetes"]
```

[67]:
```
# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify = y, test_size = 0.2, random_state=seed)
```

[68]:
```
# Train the model
Final_Model.fit(X_train, y_train);
```

This model was saved to Trained_Model.sav using the following command to power the Diabetes Predictor Tool.

[73]:
```
# Save the model for use in the application
pickle.dump(Final_Model, open('Trained_Model.sav', 'wb'))
```

Validation

The benchmark suggested for acceptance of the model by First-Class Health Care System for the precision and accuracy metrics were both 65%. The use of the final data set improved the model's accuracy and precision slightly to 71.9583 and 70.379, respectively. Therefore, the model functioned met the baseline goal for use and was accepted by First-Class Health Care System.

```
[69]: # Test the model for accuracy
print("Current accuracy score:", round(Final_Model.score(X_test, y_test)*100,4))
```

```
Current accuracy score: 71.9583
```

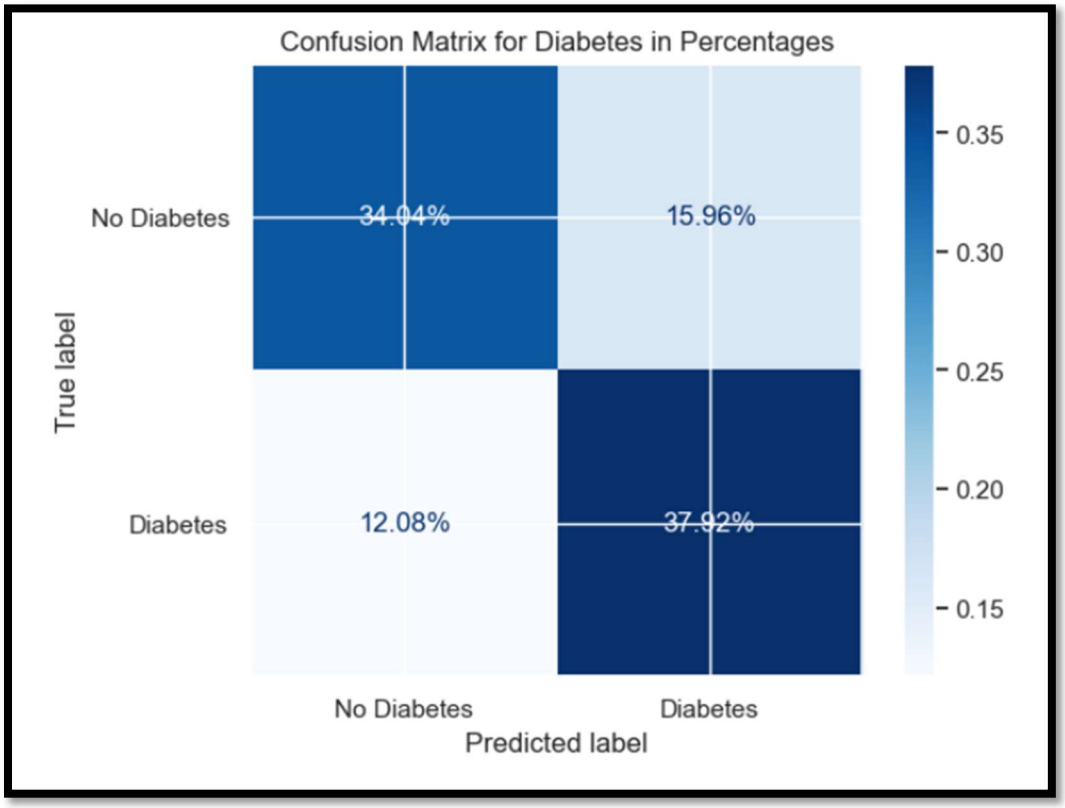
```
[70]: # Test the model for precision
print("Current precision score:", round(precision_score(y_test, Final_Model.predict(X_test))*100,4))
```

```
Current precision score: 70.379
```

Confusion Matrix

To gain a better understanding of how these metrics will affect the First-Class Health Care Systems Diabetes Predictor Tool, the following command was used to create a confusion matrix utilizing Scikit Learn,

```
[72]: # Display Confusion Matrix in percentages
confusion_matrix = metrics.confusion_matrix(actual, predicted)
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix/np.sum(confusion_matrix), display_labels = ["No Diabetes", "Diabetes"])
cm_display.plot(values_format = '.2%', cmap='Blues')
plt.title('Confusion Matrix for Diabetes in Percentages')
plt.show()
```



Patients were correctly identified as having diabetes when they had diabetes and not having diabetes when they did not have diabetes over 71% of the time. Opportunities for improvement exist with the model predicting diabetes 15.96 % of the time when the patient did not have diabetes and predicting no diabetes 12.08% when the patient had diabetes. This meant that the Diabetes Predictor Tool will miss 12.08 % of patients with diabetes while expediting services for 15.96 percent of people who do not have diabetes.

The model underlying the diabetes predictor can be improved by using First-Class Health Care Systems patient data for future analysis. Another method of improvement would be to add patients' familial data to the data as the family occurrence of diabetes plays a part in diabetes prediction (Centers for Disease Control and Prevention, 2022).

The Diabetes Predictor Tool

The Diabetes Predictor Tool was developed using the Final_Data.CSV file and Trained_Model.sav file. It was written in Ipywidgets in Jupyter Notebook to allow First-Coast Health Care Systems to evaluate the Tool. Although the tool was expected to contain 10 questions or less, 12 questions were to optimize model performance metrics.

The model uses publicly available data with identifying data removed and was created on open-source programs. The Tool does not save the data entered into the survey; therefore, no security measures or licensing fees are required. The Tool was easily converted to Python code for use in First-Coast Health Care Systems proprietary web applications and the saved data with patient personal information was secured by Veridify Security® technology.

Data from the CDC is updated annually, and the model can be updated with a small retainer fee using the methodology described herein. Jupyter Notebook, Python, and its accompanying libraries Scikit-learn, Matplotlib, NumPy, and pandas are required to maintain updates of the model.

Diabetes Predictor Tool Demonstration

The Diabetes Predictor Tool can be viewed using the Instructions provided in the Preliminary User Guide. Note that the Tool uses height and weight metrics to evaluate BMI.

Two outputs were possible from the Diabetes Predictor Tool:

1. The patient is at risk for diabetes.

   Figure 1 shows that using sample patient information with a BMI of 25 or over (i.e., being overweight), increased age, and high blood pressure produces an output of diabetes.

2. The patient is not at risk for diabetes.

   Figure 2 shows a prediction of no diabetes with sample patient information using lower age and lower BMI.

Figure 2: The patient is at risk for diabetes.

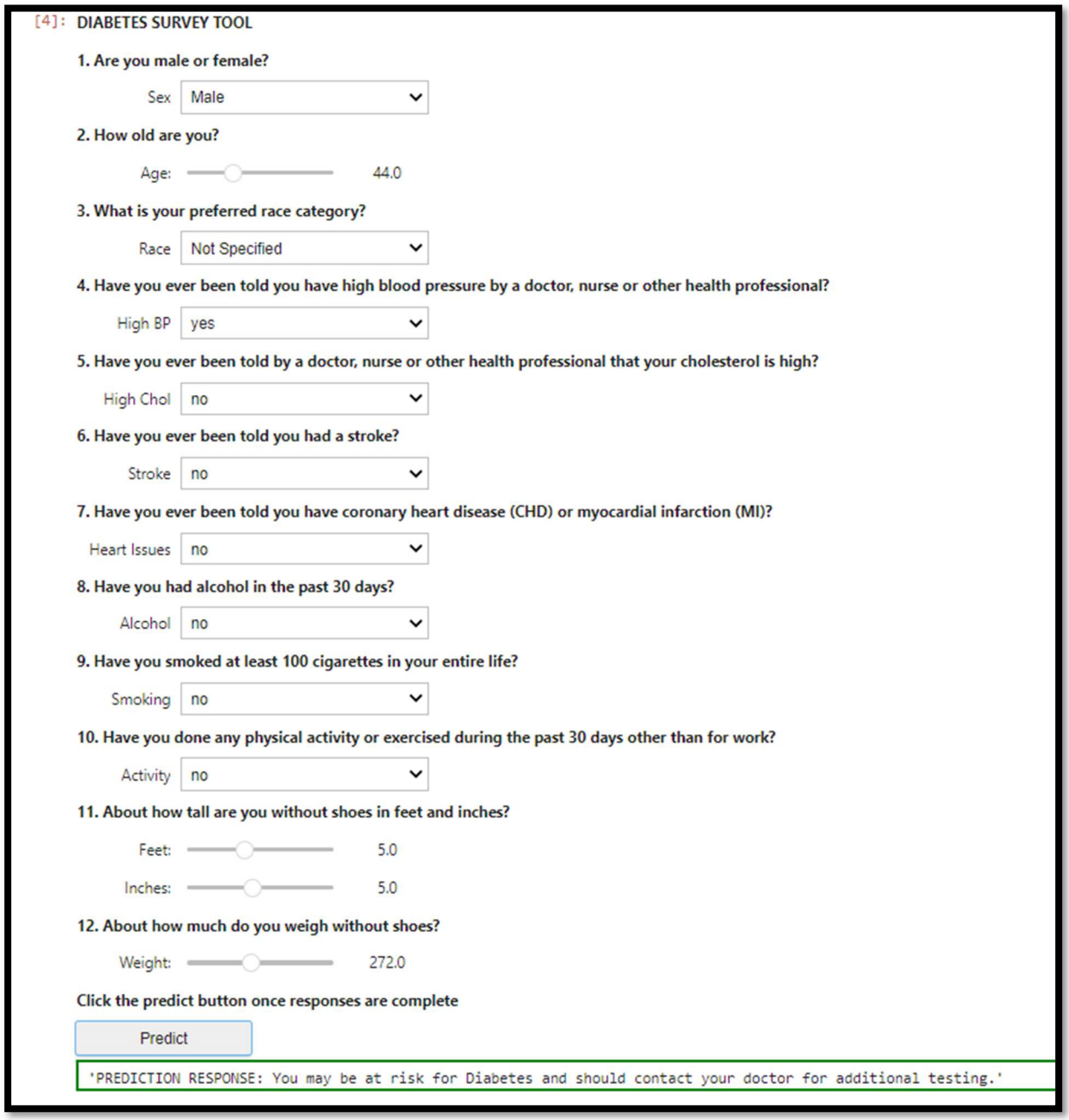
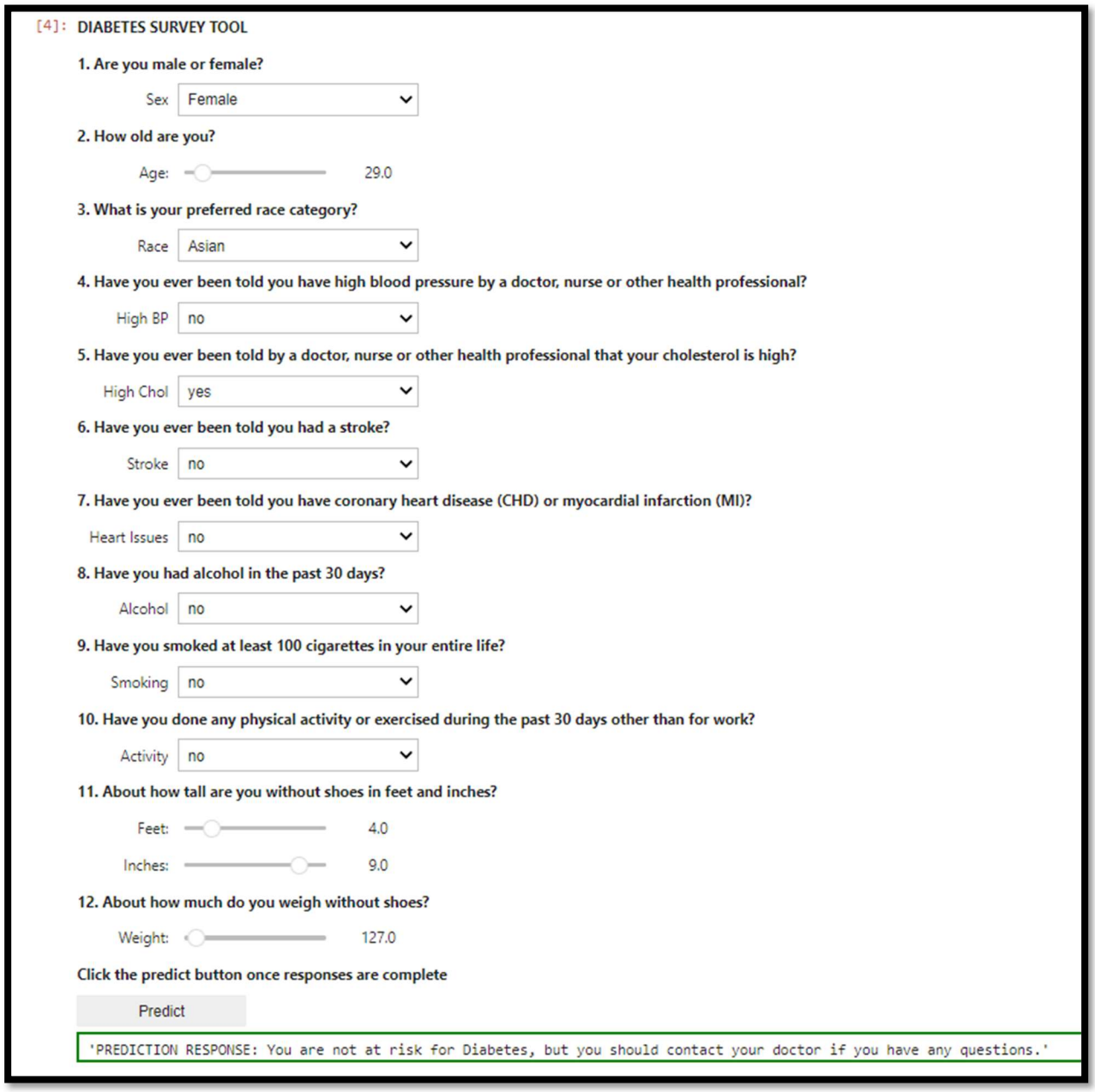


Figure 2: The patient is not at risk for diabetes.



Preliminary User Guide

The **User Guide** includes instructions to allow for the interaction with the Diabetes Predictor Tool in two formats. **Running the Application for End-Users** is suggested for lighter-weight computing environments which enables the application to run in a web browser. For data scientists with sufficient computing resources who wish to study the methodology, instructions are provided under **Running the Application and Methodology for Data Scientists**.

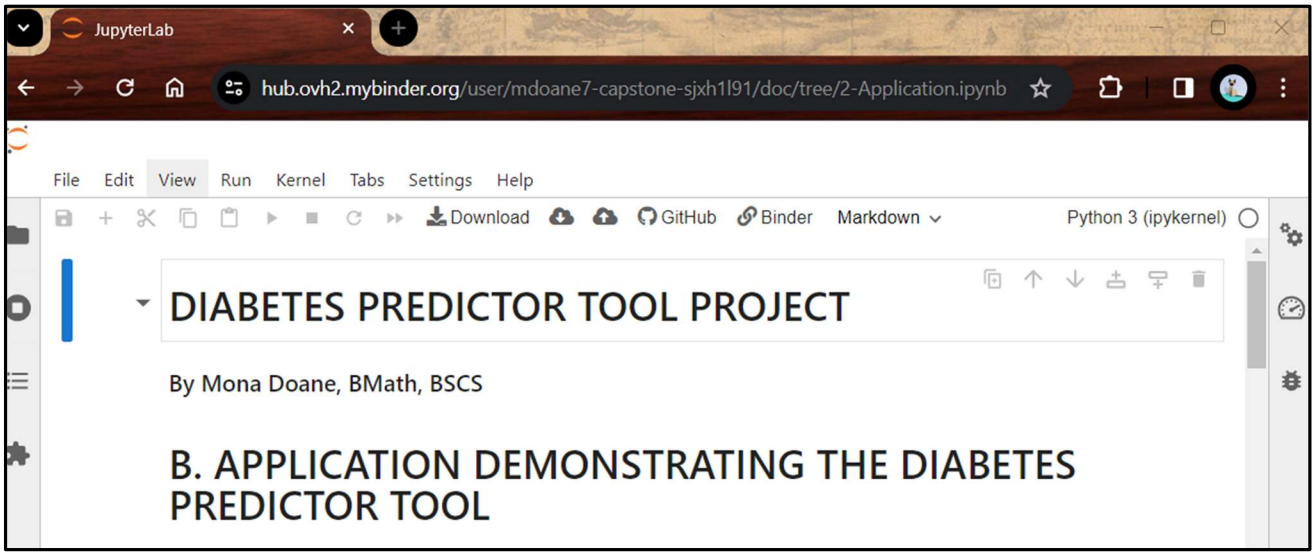
Running the Application for End-Users

Application Instructions

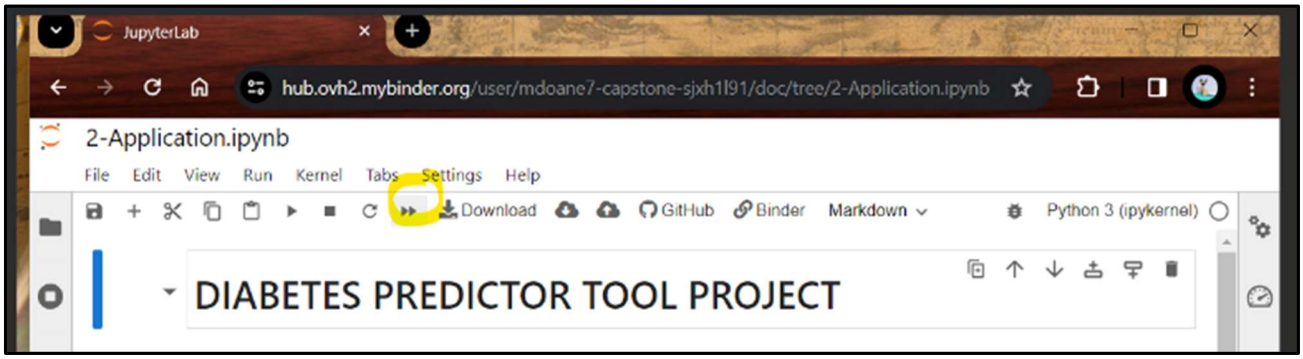
The Tool is available for use at the following location. If this link does not work on the user's computer, alternate instructions are provided as well.

https://mybinder.org/v2/gh/mdoane7/capstone/HEAD?labpath=2-Application.ipynb

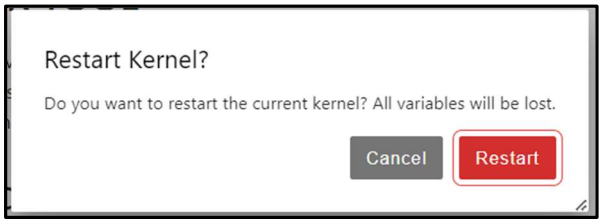
1. The Jupyter Notebook with the application will load and should look like this:



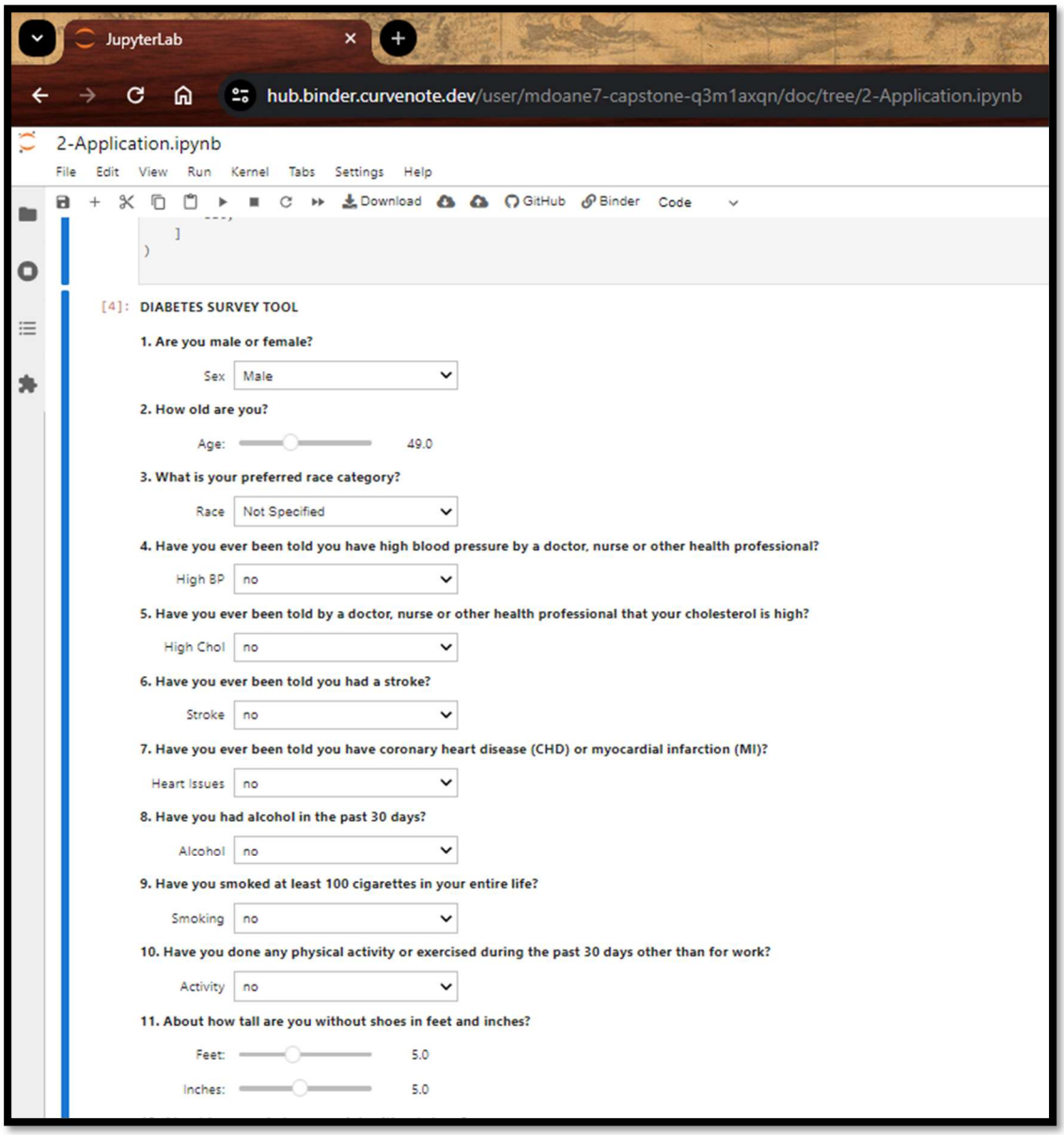
2. Click on this button (⏩) shown in yellow below to run all the cells in the notebook:



3. The following box will appear. Click Restart.



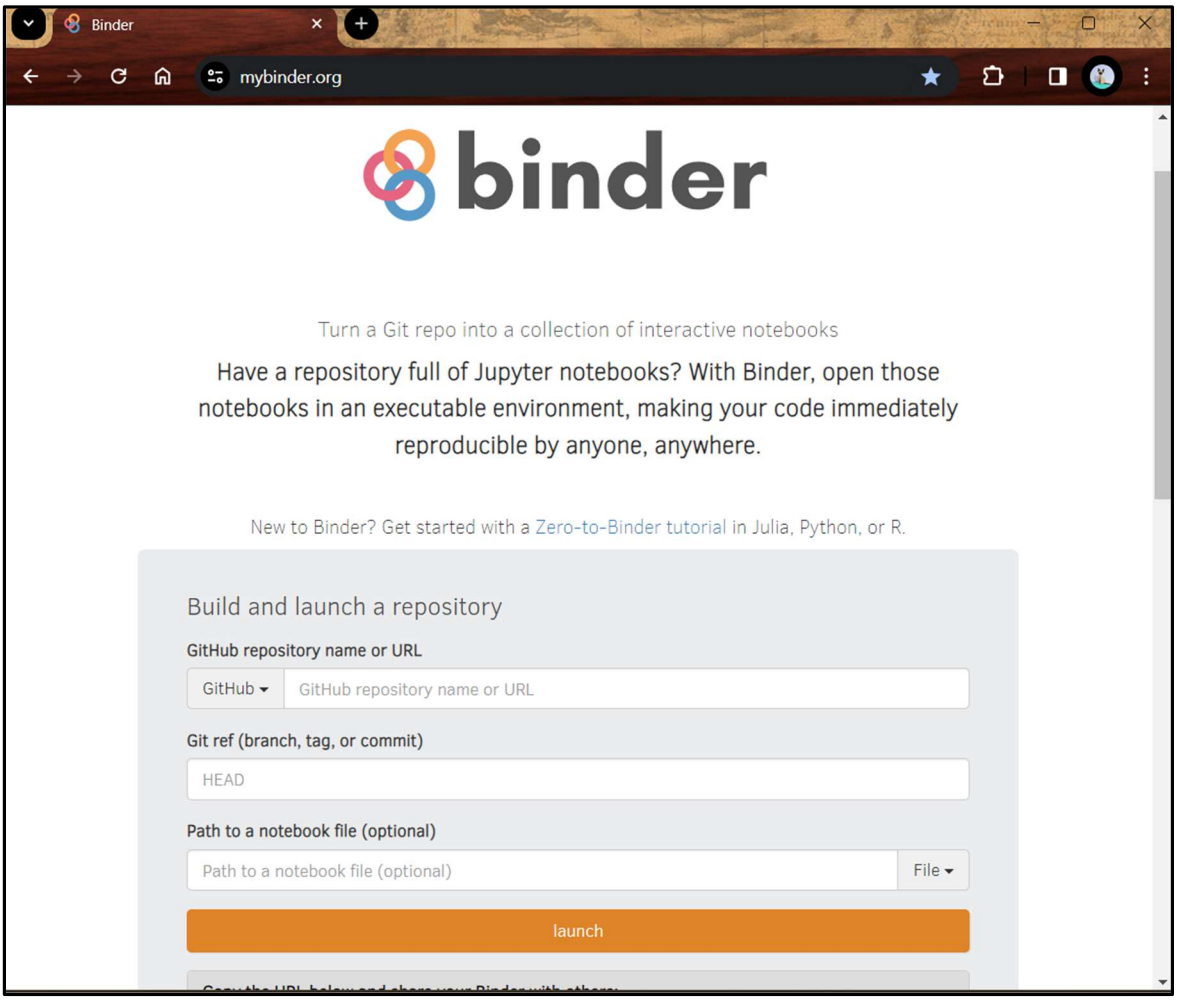
4. Scroll to the bottom of Jupyter Notebook to the second box labeled [4] with the title "DIABETES SURVEY TOOL". It should look as follows. The survey tool allows the user to enter the data and review the prediction.



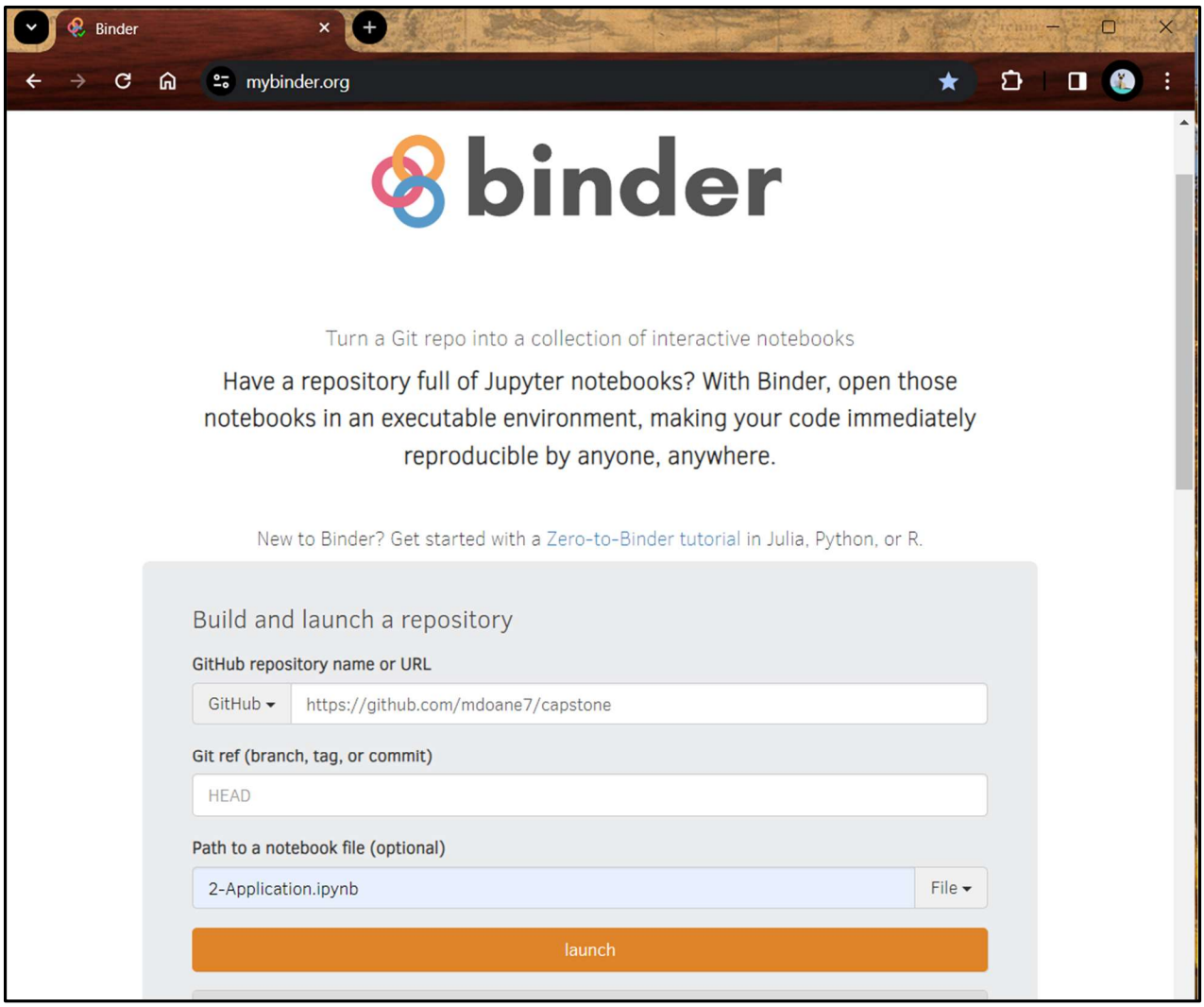
Alternate Instructions to Run Application on MyBinder.com

If the preceding does not work, instructions to use the Diabetes Predictor Tool in an interactive environment are as follows:

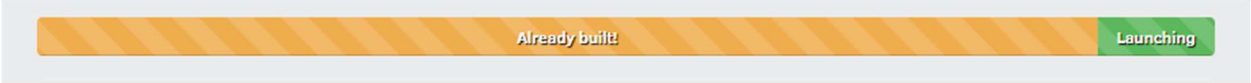
1. Open a Google Chrome window and go to https://mybinder.org/



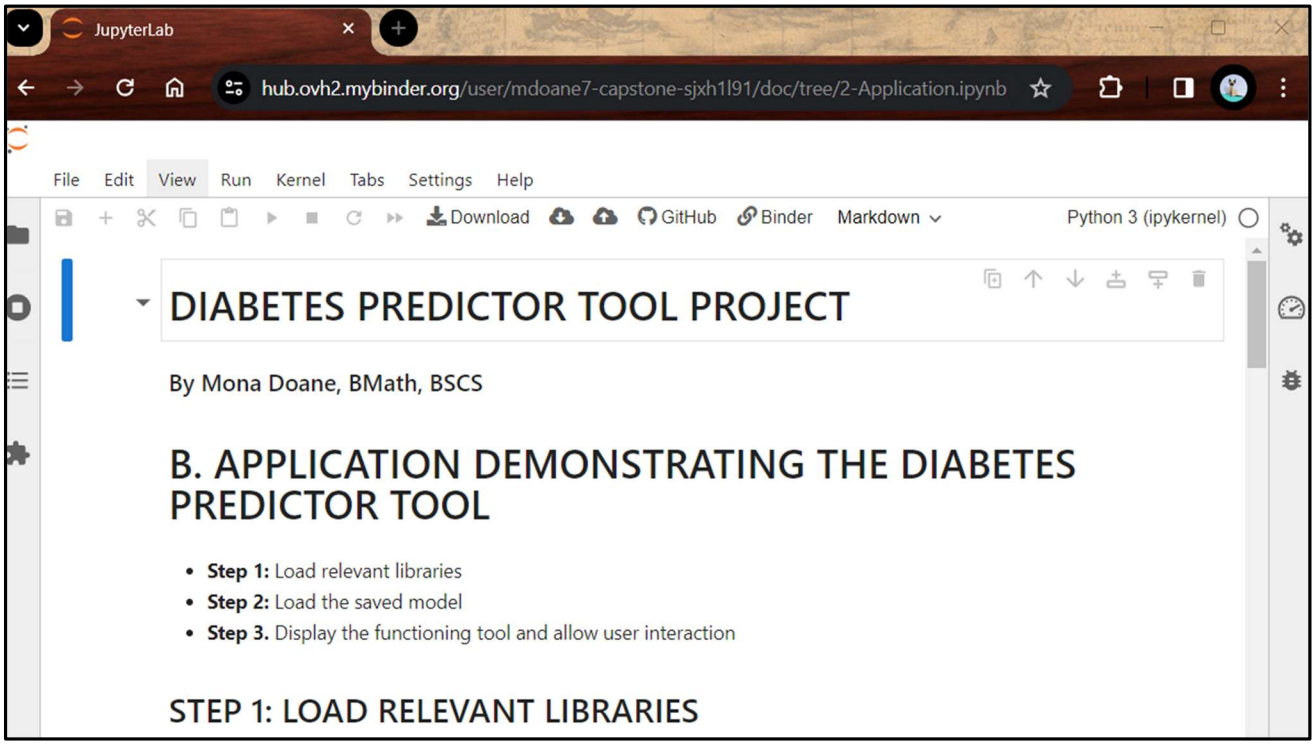
2. Complete the *Build and launch a repository* box as follows:
   a. Under the "GitHub repository name or URL" field, enter the following: https://github.com/mdoane7/capstone
   b. Under the "Path to a notebook file (optional), enter the following: 2-Application.ipynb
   c. Click the launch button.



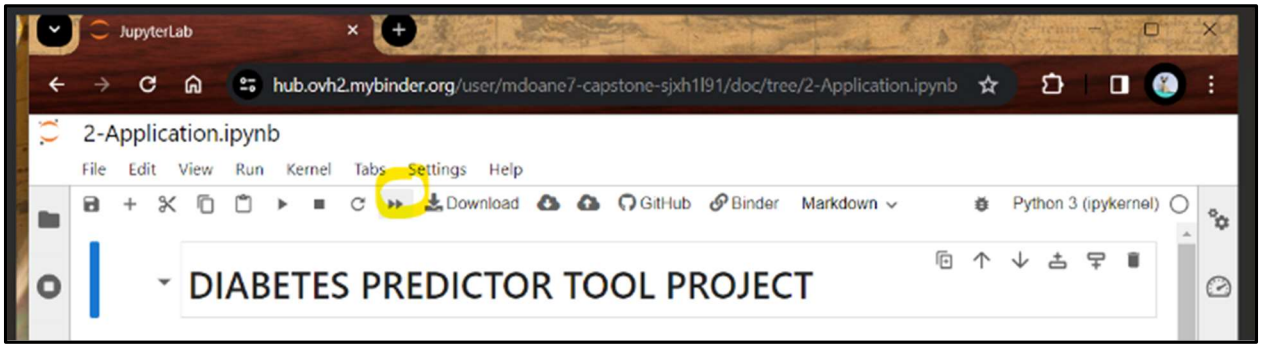
NOTE: It may take a moment for the docker image to load and run.



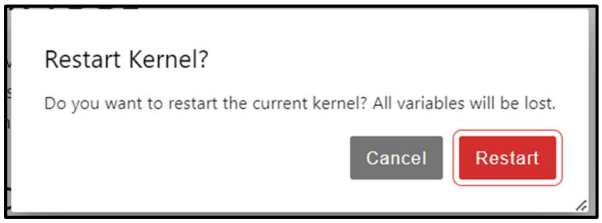
3. The Jupyter Notebook with the application will load and should look like this:



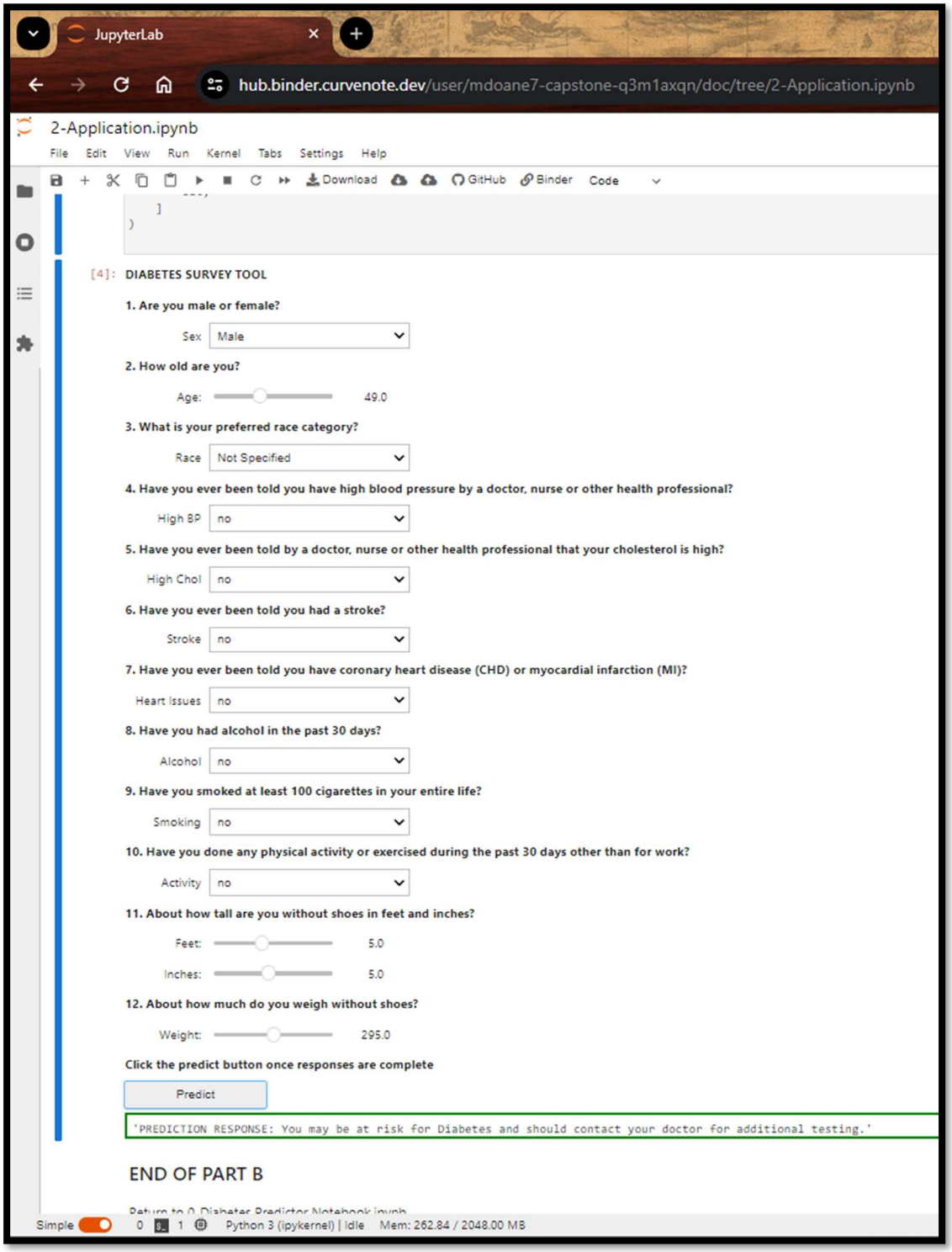
4. Click on this button (⏩) shown in yellow below to run all the cells in the notebook:



5. The following box will appear. Click Restart.



6. Scroll to the bottom of Jupyter Notebook to the second box labeled [4] with the title "DIABETES SURVEY TOOL". It should look as follows. The survey tool allows the user to enter the data and review the prediction.

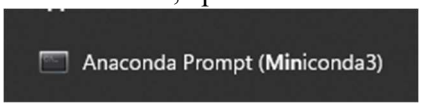


Running the Application and Methodology for Data Scientists

For data scientists who wish to run the complete analysis of the methodology used to develop the machine learning model in Jupyter Notebook, the Miniconda environment provides a compact data analysis environment complete with all the tools needed (Neagoie, 2024).

1. Install Miniconda from https://docs.conda.io/projects/miniconda/en/latest/
   a. Choose the appropriate installer based on your operating system.
   b. Download the installer and run it.
   c. Follow the instructions on-screen to complete installation. Use the recommended settings.

2. Once installed, open the Miniconda prompt from the Windows menu.



3. Once the terminal opens, type in the following. If the version appears, Miniconda has been successfully installed.

```
conda --version
```

4. Type in the following to change the directory to desktop.

```
cd desktop
```

5. To create a new project, create a folder on your desktop called diabetes_project.

```
>mkdir diabetes_project
```

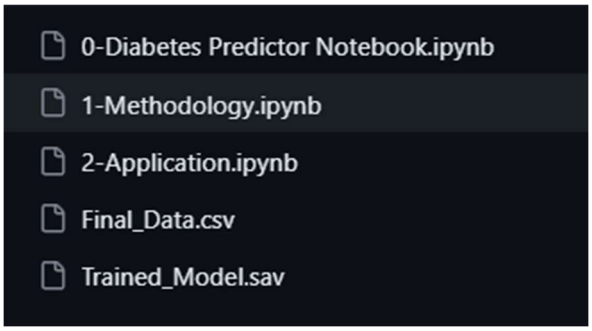
6. Change the current directory to the diabetes_project.

```
>cd diabetes_project
```

7. Create a conda environment within your new project. Press y when asked to proceed.

```
conda create --prefix ./env pandas numpy matplotlib scikit-learn
```

8. Download all source files from https://github.com/mdoane7/capstone into diabetes_project.



9. Type in the following to list all conda environments on your computer.

```
>conda env list
```

10. Copy the path of the env file under diabetes_project.

11. Activate your project as follows.

```
>conda activate \Desktop\diabetes_project\env
```

12. Once activated, install Jupyter Notebook by typing the following.

```
>conda install jupyter
```

13. Load Jupyter Notebook by typing in the following.

```
>jupyter notebook
```

14. The window should launch in your browser, and you can navigate to the Jupyter Notebooks.

15. The Home directory should contain the following:
    0-Diabetes Predictor Notebook.ipynb
    1-Methodology.ipynb
    2-Application.ipynb

16. The data file must be downloaded into the diabetes_project folder from Kaggle at the following location and saved as ***Diabetes_Dataset_2021***.
    https://www.kaggle.com/datasets/dariushbahrami/cdc-brfss-survey-2021

17. The Methodology file contains the complete methodology used to establish the model used for the Diabetes Predictor Tool.

References

Ali, M. (2022, August 5). *Supervised Machine Learning*. Retrieved from datacamp.com: https://www.datacamp.com/blog/supervised-machine-learning

Centers for Disease Control and Prevention. (2022, April 5). *Diabetes Risk Factors*. Retrieved from Centers for Disease Control and Prevention: https://www.cdc.gov/diabetes/basics/risk-factors.html

Centers for Disease Control and Prevention. (2023, September 5). *What is diabetes?* Retrieved from cdc.gov: https://www.cdc.gov/diabetes/basics/diabetes.html

Fregoso-Aparicio, L. N. (2021, December 20). *Machine learning and deep learning predictive models for type 2 diabetes: a systematic review.* Retrieved from dmsjournal.biomedcentral.com: https://dmsjournal.biomedcentral.com/articles/10.1186/s13098-021-00767-9

Hotz, N. (2023, January 19). *What is CRISP DM?* Retrieved from datascience.pm.com: https://www.datascience-pm.com/crisp-dm-2/

Hussein, W. M. (2022, December 05). Identifying risk factors associated with type 2 diabetes based on data analysis. *Measurement: Sensors*, p. https://www.sciencedirect.com/science/article/pii/S2665917422001775?via%3Dihub.

IBM Corporation. (2021, August 17). *How SVM Works*. Retrieved from IBM.com: https://www.ibm.com/docs/en/spss-modeler/saas?topic=models-how-svm-works

Laoyan, S. (2022, October 15). *What is agile methodology? (a beginner's guide)*. Retrieved from asana.com: https://asana.com/resources/agile-methodology

Neagoie, A. (2024, January 4). *Complete A.I. & Machine Learning, data science bootcamp.* Retrieved from udemy.com: https://www.udemy.com/course/complete-machine-learning-and-data-science-zero-to-mastery/

Shung, K. (2018, March 15). *Accuracy, Precision, Recall or F1?* Retrieved from towardsdatascience.com: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Sujan, N. (2018, November 22). *Top 5 Machine Learning Libraries in Python*. Retrieved from towardsdatascience.com: https://towardsdatascience.com/top-5-machine-learning-libraries-in-python-e36e3e0e02af

U.S. Centers for Medicare & Medicaid Services. (2023, September 6). *Behavioral risk factor surveillance system*. Retrieved from CMS.gov: https://www.cms.gov/about-cms/agency-information/omh/resource-center/hcps-and-researchers/data-tools/sgm-clearinghouse/brfss

Yu, W. L. (2009, December 18). *Application of support vector machine modeling for prediction of common diseases: the case of diabetes and pre-diabetes.* Retrieved from BMC Medical Informatics and Decision Making: https://doi.org/10.1186/1472-6947-10-16