

## **C964 Computer Science Capstone**

Anthony J. Coots

College of Information Technology, Computer Science, Western Governors University

C964: Computer Science Capstone

September 4<sup>th</sup>, 2023

Table of Contents

---

<b>Letter of Transmittal</b>	<b>PG 3 – 5</b>
<b>Executive Summary</b>	<b>PG 6 – 8</b>
<b>Documentation</b>	<b>PG 9 – 37</b>
<b>References</b>	<b>PG 38</b>

## Letter of Transmittal

---

John Smith

Director of Western General Medical Group

9001 ABC Drive, Los Angeles, CA 90001

September 9<sup>th</sup>, 2023

Dear Dr. Smith,

I am writing to you with great enthusiasm and a common goal in mind—to enhance the detection of diabetes among patients within Western General Medical Group. Over the past few decades, the medical community has witnessed a significant increase in the prevalence of diabetes, making early detection and management crucial. As someone who understands the challenges faced by patients and medical practitioners, I would like to propose the development of a data reporting tool for preliminary diabetes detection.

This standalone data tool will benefit both patients and practice by streamlining the process of identifying potential diabetes cases. It will operate by analyzing three key factors that are closely associated with diabetes: Body Mass Index (BMI), family Diabetes pedigree score, and age. Importantly, this tool will operate locally within the medical practice, ensuring the utmost security of patient data against any potential vulnerabilities.

The tool will be designed to read data from a locally stored file and then undergo training based on the data within that file. By doing so, it will be able to detect the presence of diabetes with an impressive accuracy rate of around 70-80%, providing valuable assistance in the early

detection of this chronic condition. This level of predictive accuracy promises to be a valuable asset in general efforts to improve patient care.

To ensure the integrity of both the tool and practice, the reporting tool will read data from a file while respecting patient privacy. It will specifically identify the columns related to BMI, diabetes pedigree, and age, without associating this information of any individual patient. This approach will ensure full compliance with patient privacy regulations, including PII, PHI, and HIPAA rights.

Implementing this solution will greatly enhance the practice's ability to conduct simple yet effective health assessments. It will address critical factors that are often overlooked or underemphasized in patient care, thereby contributing to the early detection of diabetes. The required funding for this project is reasonable, with the project divided into two stages:

1. Stage One will involve setting up the practice framework, which includes acquiring a Windows 10 machine capable of moderate processing, roughly \$1,000. The tool's environment will be free to use, ensuring cost-efficiency.
2. Stage Two will encompass the data product as a whole, estimated to require approximately 180 work hours, with a cost of \$7,500. This budget will cover all aspects of planning, design, development, and documentation bringing the final total to roughly \$9,000.

I want to assure you that there will be no rush for approval, and I will welcome open communication on this proposal. As a Bachelor of Science in Computer Science and a seasoned database administrator, I will be well-equipped to lead this project. However, what will truly drive this data product is my personal experience as a type 1 diabetic, which will underscore the significance of early detection and effective management.

I will be eager to discuss this proposal further and address any questions or concerns you may have. Please feel free to reach out to me at [anthonyjcoots@yahoo.com](mailto:anthonyjcoots@yahoo.com). Your feedback and insights will be invaluable, and I look forward to the opportunity to work together in improving patient care within Western General Medical Group.

Thank you for your time and consideration.

Sincerely,

Anthony Coots.

A handwritten signature in black ink, appearing to read 'Anthony Coots', with a stylized, cursive script.

## Executive Summary

---

IT Department

Western General Medical Group, IT Department

9001 ABC Drive, Los Angeles, CA 90001

September 9<sup>th</sup>, 2023

Dear IT Director(s),

In light of ongoing advancements in medical research and technology, there is a pressing opportunity to enhance the methodologies employed for the detection of diabetes. This executive summary presents a forthcoming solution that underscores the potential benefits of amalgamating machine learning solutions with health-related detection technologies. As an illustrative example, this proposal endorses the utilization of specific inputs such as Body Mass Index (BMI), Family Diabetes pedigree score, and age to predict the presence of diabetes.

The applicability of this proposed solution is for but not limited to Western General Medical Group to encompass any medical entity contemplating the implementation of a machine-learning solution for diabetes detection. The core objective of this product is to contribute to the early identification of diabetes by leveraging non-identifying patient data. Specifically, it will rely on the data fields representing BMI, diabetic pedigree score, and age, sourced from a CSV file, and subsequently processed into an array for data science application.

The initial dataset chosen for demonstration purposes will be obtained from Kaggle.com. This dataset contains diverse information about non-identifiable patients, both with and without

diabetes. However, it is imperative to emphasize that only BMI, pedigree score, and age data fields will be employed. This strict data selection ensures that patient identities remain confidential and that the product adheres to the highest standards of patient data privacy.

The development process of this solution will be grounded in the Agile methodology. It is important to acknowledge that while the prediction of diabetes can never be rendered with absolute certainty, ongoing improvements, and adaptations will be necessary for upholding the integrity of the tool. The active involvement of stakeholders in this continuous improvement process is critical. Unlike static products, a dynamic machine-learning tool of this nature endeavors to seamlessly blend machine learning with preliminary health-related detection technologies. In line with this, the deliverables will encompass comprehensive development documentation, a well-crafted product backlog, and an installation-to-operation guide for smooth integration.

The implementation plan comprises two pivotal stages: development and installation. The development phase will be through Spyder (Python 3.11) within the Anaconda Navigator environment. A series of stakeholder meetings will occur to assess the tool's capabilities and ensure alignment with the requirements set by the practice. Subsequently, the requirement responsibility will rest on 'Western General Medical Group' to acquire a Windows 10 machine with moderate-to-heavy processing capacity. Following this, the Spyder environment, with the Anaconda Navigator installation, will be deployed, and the Python script will be put into action once the requisite payment is delivered.

Validation and verification of the developed product will rely on a statement of work generated by the practice, accompanied by an agile framework that facilitates continuous

feedback from stakeholders. Verification will involve a detailed comparison between the data product and the predetermined requirements and objectives.

It is pertinent to highlight that all programming environments selected for this initiative are cost-free. Anaconda Navigator offers a complimentary version of Spyder (Python 3.11), which is compatible with all Windows 10 systems. Project costs are derived from critical phases such as planning, design, development, and documentation.

**Projected Timeline:**

- **Planning and Design:** 10/01/2023-10/08/2023
- **Development:** 10/08/2023-10/22/2023
- **Documentation:** 10/22/2023-11/01/2023

**Milestones:**

- **Final Design:** 10/22/2023
- **Implementation:** 11/08/2023
- **Final Release (Stop Development):** 12/31/2023

The development of this tool will be executed as a solo endeavor, requiring no additional dependencies or human resources beyond an estimated 170 work hours, at an estimated cost of \$7,500.

I look forward to engaging in further discussions regarding this proposal, and I warmly invite you to reach out to [anthonyjcoots@yahoo.com](mailto:anthonyjcoots@yahoo.com) for any queries, clarifications, or feedback.

Sincerely,

Anthony Coots.

A handwritten signature in black ink, appearing to read 'Anthony Coots', with a stylized, flowing script.



## Documentation

---

The business requirements and vision for this project necessitate the acquisition of a Windows 10 machine equipped with moderate-to-heavy processing capabilities. This hardware requirement is integral to accommodate the volume of data entries that the predictive model will utilize to make determinations regarding the presence or absence of Diabetes.

Before data product implementation, the designated machine must be configured with the requisite (free) software environments. These environments are essential to ensure the smooth execution of the project's data processing and analysis tasks. Furthermore, the successful development and deployment of the project are contingent upon the upfront payment of \$9,000. This financial investment is crucial to facilitate the project's progression, planning, design, development, and deployment stages.

This strategic approach aligns with the project's overarching vision, which aims to harness technology and data-driven methodologies to enhance the practice's capability for early detection of Diabetes, ultimately contributing to improved patient care and healthcare outcomes.

The raw data used in this project has limited application and thus requires manual cleaning to ensure its suitability for analysis. Specifically, a few pre-existing columns required removal from the dataset from <https://www.kaggle.com/datasets/nanditapore/healthcare-diabetes>.

**FIGURE A.1**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Id	Pregnanci	Glucose	BloodPres	SkinThickr	Insulin	BMI	DiabetesF	Age	Outcome				
2	1	6	148	72	35	0	33.6	0.627	50	1				
3	2	1	85	66	29	0	26.6	0.351	31	0				
4	3	8	183	64	0	0	23.3	0.672	32	1				
5	4	1	89	66	23	94	28.1	0.167	21	0				
6	5	0	137	40	35	168	43.1	2.288	33	1				
7	6	5	116	74	0	0	25.6	0.201	30	0				
8	7	3	78	50	32	88	31	0.248	26	1				
9	8	10	115	0	0	0	35.3	0.134	29	0				
10	9	2	197	70	45	543	30.5	0.158	53	1				
11	10	8	125	96	0	0	0	0.232	54	1				
12	11	4	110	92	0	0	37.6	0.191	30	0				
13	12	10	168	74	0	0	38	0.537	34	1				
14	13	10	139	80	0	0	27.1	1.441	57	0				
15	14	1	189	60	23	846	30.1	0.398	59	1				
16	15	5	166	72	19	175	25.8	0.587	51	1				
17	16	7	100	0	0	0	30	0.484	32	1				
18	17	0	118	84	47	230	45.8	0.551	31	1				
19	18	7	107	74	0	0	29.6	0.254	31	1				
20	19	1	103	30	38	83	43.3	0.183	33	0				
21	20	1	103	30	38	83	43.3	0.183	33	0				

“Figure A.1” displays the initial dataset acquired from the provided link. Subsequently, the data underwent a cleaning process to transform it into the format depicted in “Figure A.2.”

**FIGURE A.2**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	33.6	0.627	50	1										
2	26.6	0.351	31	0										
3	23.3	0.672	32	1										
4	28.1	0.167	21	0										
5	43.1	2.288	33	1										
6	25.6	0.201	30	0										
7	31	0.248	26	1										
8	35.3	0.134	29	0										
9	30.5	0.158	53	1										
10	0	0.232	54	1										
11	37.6	0.191	30	0										
12	38	0.537	34	1										
13	27.1	1.441	57	0										
14	30.1	0.398	59	1										
15	25.8	0.587	51	1										
16	30	0.484	32	1										
17	45.8	0.551	31	1										
18	29.6	0.254	31	1										
19	43.3	0.183	33	0										
20	34.6	0.529	32	1										
21	30.2	0.204	27	0										

The limited applicability of the Kaggle dataset arises from the fact that the tool relies on data that the medical practice is responsible for collecting and maintaining. The dataset downloaded from Kaggle was solely utilized for demonstration, training, and development purposes. The cleaned data used during the training and development phase is illustrated in “Figure A.3.”

**FIGURE A.3**

Index	BMI	Pedigree	Age	Outcome
0	33.6	0.627	50	1
1	26.6	0.351	31	0
2	23.3	0.672	32	1
3	28.1	0.167	21	0
4	43.1	2.288	33	1
5	25.6	0.201	30	0
6	31	0.248	26	1
7	35.3	0.134	29	0
8	30.5	0.158	53	1
9	0	0.232	54	1
10	37.6	0.191	30	0
11	38	0.537	34	1
12	27.1	1.441	57	0

“Figure A.3” provides a visual representation of the data for later analysis techniques employed within this capstone project. The figure showcases the dataset's transformation into a data frame format, a two-dimensional data structure natively supported by the Python library 'pandas.' This data frame serves as an essential organizational tool, like that of a dictionary, for the dataset, enabling efficient mapping of data rows, each representing a unique patient entry, to their corresponding data columns. These columns encompass critical patient attributes such as Body Mass Index (BMI), Diabetes pedigree score, and age. The strategic use of the data frame not only enhances data clarity but also facilitates subsequent data analysis and modeling.

In the second stage of the analysis, both descriptive and predictive analytics are generated. The 'sklearn' library facilitates this process through machine learning and data analysis. Specifically, the data is subjected to training a supported vector machine (SVM) model, imported from the 'sklearn' library and denoted as 'svm.' This supervised learning approach enables the model to learn and establish patterns within the dataset, making it adept at making predictions based on new, unseen data. The SVM model, recognized for its versatility and effectiveness in various domains, plays a pivotal role in extracting meaningful insights from the dataset, thereby enabling informed decision-making and prediction of diabetes presence.

The processes demonstrated in the figures below, attached to the data frame identified in “Figure A.3” exemplify the significance of data preprocessing and model training in this project. By effectively transforming the raw dataset into a structured data frame format, the project ensures the efficient organization and retrieval of relevant patient information. Subsequent analysis, powered by the 'sklearn' library, empowers the project to deliver both descriptive insights, which aid in understanding dataset characteristics, and predictive analytics, which enable the identification of potential diabetes cases. This approach underscores the project's commitment to data-driven decision-making and its potential to contribute valuable insights to the field of healthcare and diabetes detection. Below is the model implementation and use of the ‘sklearn’ library as mentioned.

**FIGURE B.1.1** – Model Initialization and Data Preparation

```
29  
30     modelToTrain = svm.SVC()  
31     y = dataframe.values[:, 3]  
32     X = dataframe.values[:, 0:3]  
33
```

“Figure *B.1.1*” illustrates the code for initializing the supported vector machines (SVM) model within the project. A critical step in this process involves defining the 'modelToTrain' variable configured as a function call to the SVM classification function within the 'svm' import. Additionally, two essential variables, 'X' and 'y,' are declared as array objects. These variables are pivotal in preparing the dataset for subsequent model training.

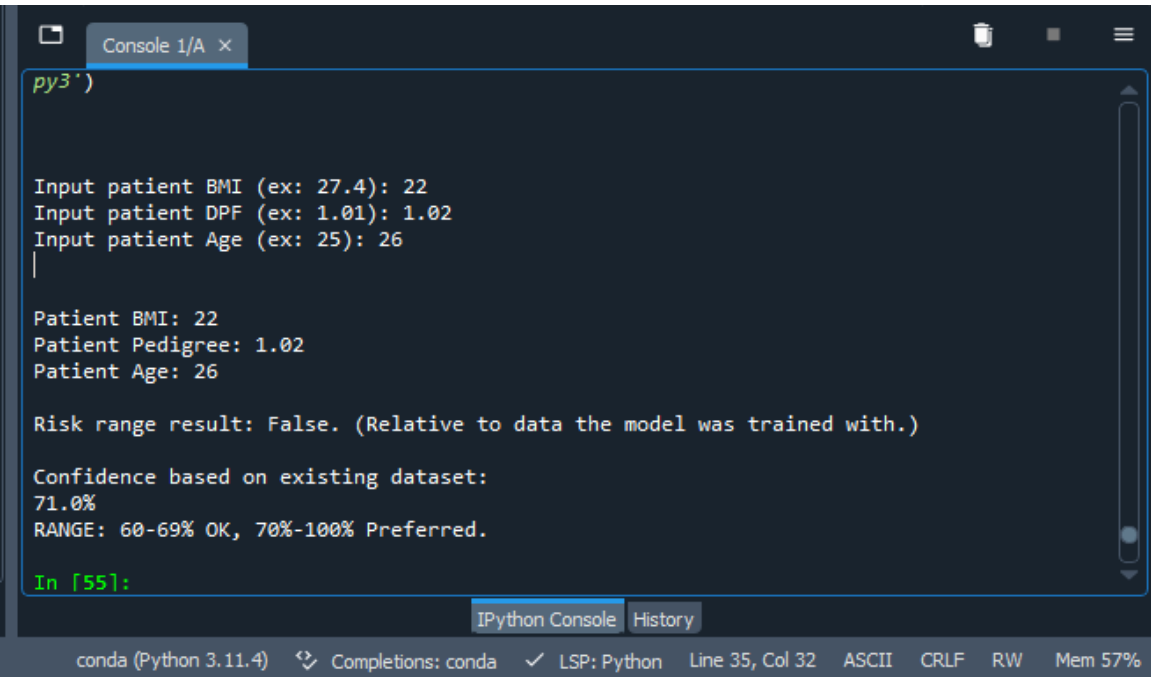
**FIGURE B.1.2** – Predictive Modeling and Outcome Generation

```

69 #####
70 # Application to algorithm, training algorithm, model creation.
71 # Inspiration from Dr. Jim Ashe, see link: https://wgu.webex.com/recording/service/sites/wgu/recording/9469df1f7d63103abf7f0050568114a0/playback
72 X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size = 0.3)
73
74 modelToTrain.fit(X_train, y_train)
75
76 y_pred = modelToTrain.predict(X_test)
77
78 patBMI = input("\n\nInput patient BMI (ex: 27.4): ")
79 patDPF = input("Input patient DPF (ex: 1.01): ")
80 patAge = input("Input patient Age (ex: 25): ")
81
82 # Procedure for user to apply model, use IPython Console in Spyder.
83 print("\n")
84 print("Patient BMI: " + patBMI)
85 print("Patient Pedigree: " + patDPF)
86 print("Patient Age: " + patAge + "\n")
87
88 # Application of the model to single input.
89 varResult = str(modelToTrain.predict([[patBMI, patDPF, patAge]]))
90
91 if(varResult == '[0.]'):
92     varResult = "False."
93 else:
94     varResult = "True."
95
96 print("Risk range result: " + varResult + " (Relative to data the model was trained with.)" + "\n")
97 print("Confidence based on existing dataset: ")
98 print(str(round(float(metrics.accuracy_score(y_test, y_pred)), 2) * 100) + "%")
99 print("RANGE: 60-69% OK, 70%-100% Preferred.")
100 #####
101

```

“Figure B.1.2” drafts the predictive method of the project. The code snippet demonstrates how the model analyzes the dataset and generates predictions relative to the dataset it has been trained with. In particular, the 'modelToTrain.predict(...)' function is utilized to make predictions based on the knowledge acquired during the 'modelToTrain.fit(...)' training process. The results of these predictions are then stored in the 'varResult' variable, which subsequently prints a 'True' or 'False' statement to the console, signifying the model's assessment of the presence or absence of Diabetes. For further insights, please refer to “Figure B.1.3.”

**FIGURE B.1.3** – Console Interaction for Prediction Analysis

```
py3')

Input patient BMI (ex: 27.4): 22
Input patient DPF (ex: 1.01): 1.02
Input patient Age (ex: 25): 26
|

Patient BMI: 22
Patient Pedigree: 1.02
Patient Age: 26

Risk range result: False. (Relative to data the model was trained with.)

Confidence based on existing dataset:
71.0%
RANGE: 60-69% OK, 70%-100% Preferred.

In [55]:
```

The screenshot shows a Jupyter Notebook interface with a terminal window titled 'Console 1/A'. The terminal displays a Python script execution. It prompts for patient BMI, DPF, and Age, which are entered as 22, 1.02, and 26 respectively. The output shows the patient's data and a risk range result of False, along with a confidence level of 71.0% and a range of 60-69% OK, 70%-100% Preferred. The terminal also shows the current line and column (Line 35, Col 32) and memory usage (Mem 57%).

“Figure B.1.3” provides a visualization of the console interaction, which describes the model's predictions in relation to the input analysis and in comparison to the dataset maintained by the medical practice, in this instance the CSV downloaded at the beginning of the code. This output offers valuable insights into the model's performance and its' capacity to assess the likelihood of Diabetes presence in analyzed data.

These code segments collectively contribute to the project's objective of harnessing machine learning techniques to enhance the early detection of Diabetes, thereby improving patient care and healthcare outcomes.

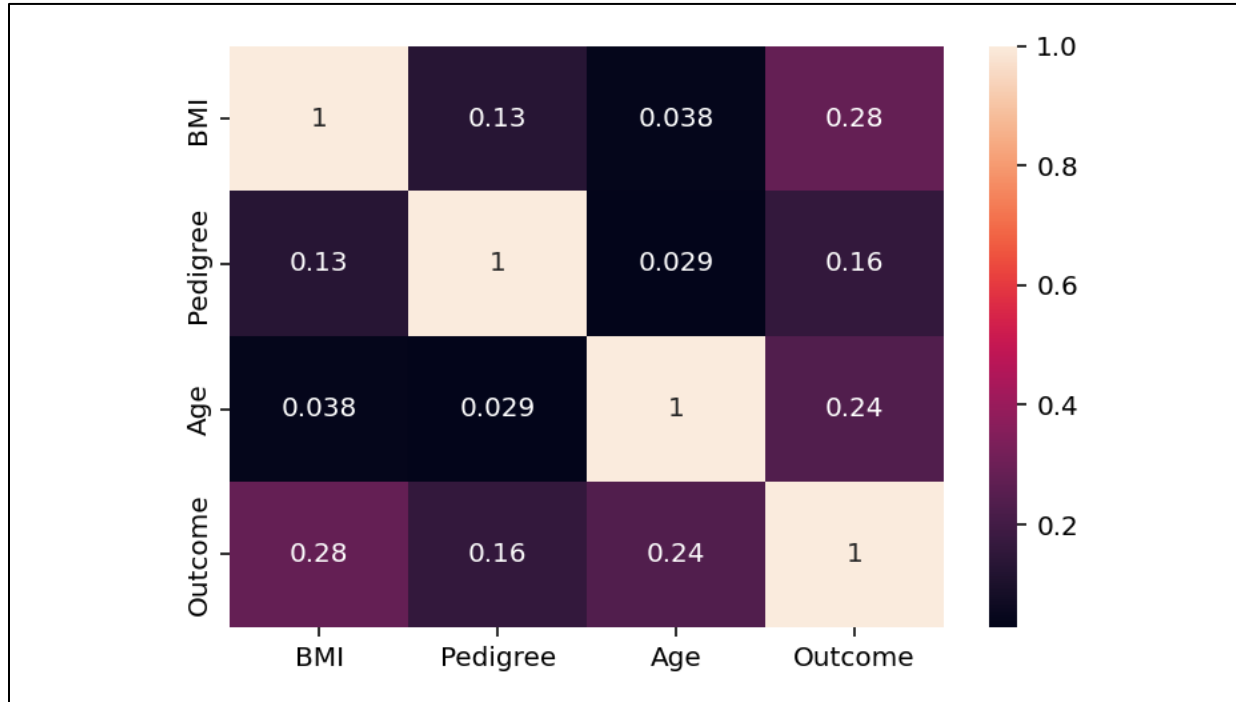


### Hypothesis:

The hypothesis of this data product sets the creation of a machine learning tool designed to predict the presence of Diabetes based on variables encompassing Body Mass Index (BMI), Diabetes pedigree score, and age, achieving a level of accuracy classed as "good."

Formally, "The data product achieving "good" accuracy defined as achieving a threshold of 70% or higher, will effectively classify Diabetes presence based on three key factors, BMI, Diabetes pedigree score, and age."

This definition is in recognition of the complex nature of Diabetes, which occurs across diverse demographic groups with varying factors. While many machine learning applications aim for a 90% accuracy rate, achieving such precision in the context of Diabetes prediction typically requires a detailed team with advanced expertise in both data science and biology. The current performance of the tool consistently attains roughly or exceeds the 70% accuracy threshold, suggesting that it aligns with and fulfills the stated objectives and requirements effectively.

**FIGURE C.1.1** – Heatmap/Correlation Matrix

“Figure C.1.1” presents a heatmap depicting the correlation matrix of the dataset. The Python code snippet below demonstrates the steps involved in creating this visualization. Notably, the code reads the dataset, initializes a support vector machine (SVM) model, and generates the heatmap with annotations.

In “Figure C.1.1”, the heatmap provides insights into the correlations between the individual dataset variables and Diabetes presence, demonstrating no two variables may provide immediate association, though will provide emphasis on why all three need to be considered in later visualizations. “Figure C.1.2” introduces a 3D scatterplot that visually represents the relationships among BMI, Diabetes pedigree score, age, and Diabetes outcomes. However, to improve interpretability, “Figure C.1.3” employs a more user-friendly visualization, depicting the

distribution of weights within the dataset. This figure demonstrates how the machine learning model's predictions are influenced by varying combinations of BMI and Pedigree, offering a clearer understanding of the model's decision-making process.

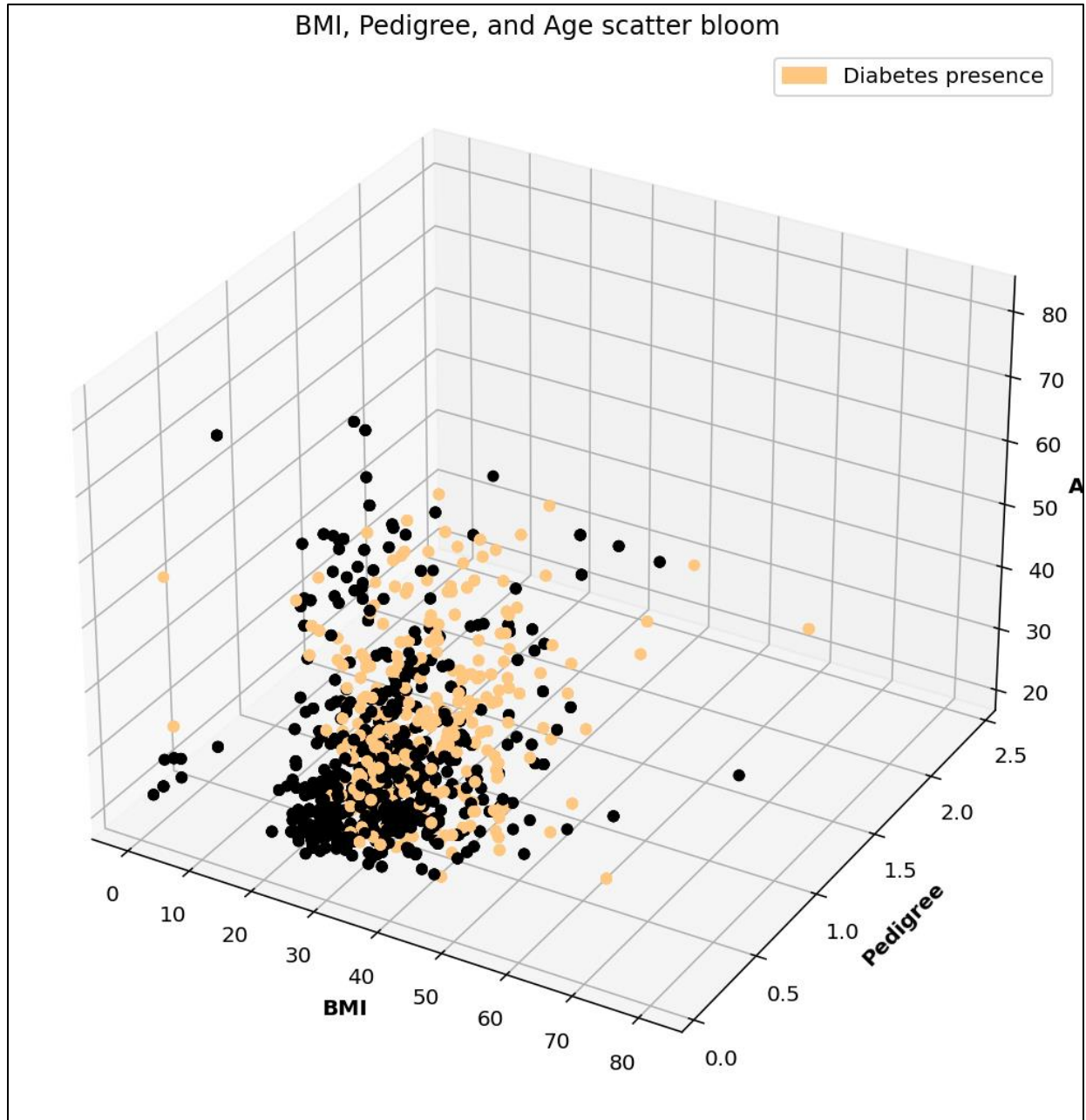
Code:

```
dataFrame = pd.read_csv(url, names=dataFields)

modelToTrain = svm.SVC()

sb.heatmap(dataFrame.corr(), annot=True)

plt.show()
```

**FIGURE C.1.2** – 3D Scatterplot.

“Figure C.1.2” showcases a 3D scatterplot that visually represents the relationships among three variables: BMI, Diabetes pedigree score (DPF), and age. The Python code below illustrates the steps taken to create this interactive visualization.

Code:

```
fig = plt.figure(figsize=(16, 9))

ax = plt.axes(projection='3d')

# Grid

ax.grid(b=True, color='grey', linestyle='-.', linewidth=0.3, alpha=0.2)

# Plot

plot3d = ax.scatter3D(varBMI, varDPF, varAge, alpha=1, c=varOutcome,
cmap=plt.get_cmap('copper'))

plt.title('BMI, Pedigree, and Age Scatterplot')

ax.set_xlabel('BMI', fontweight='bold')

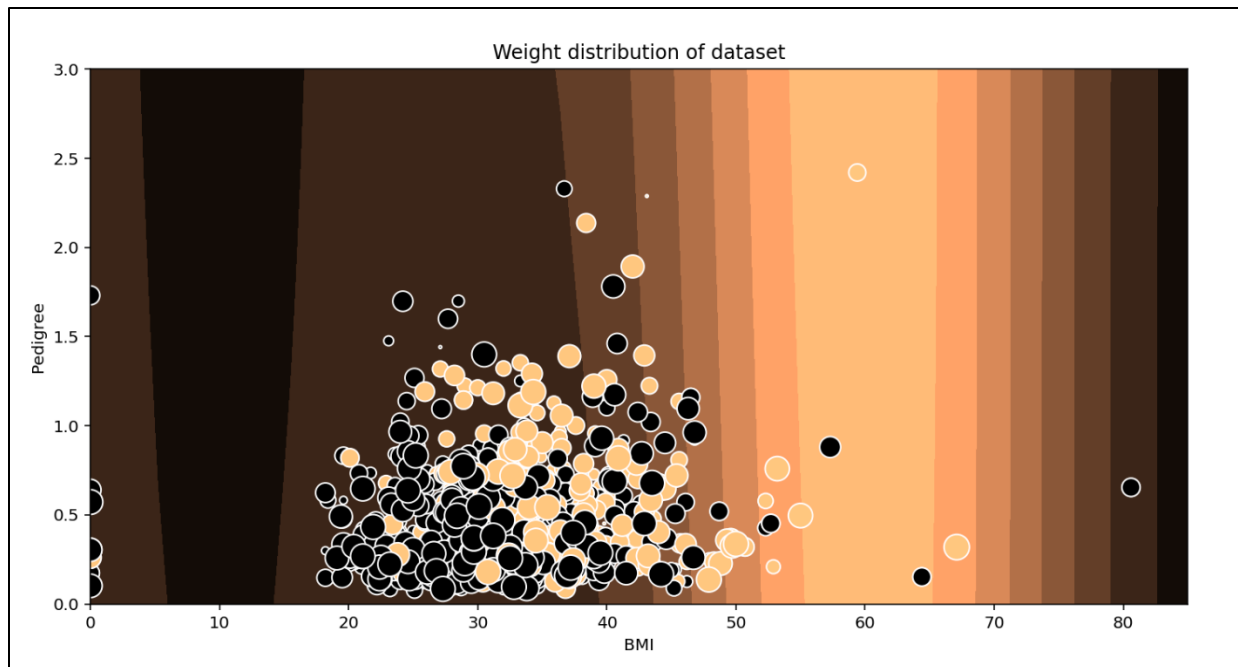
ax.set_ylabel('Pedigree', fontweight='bold')

ax.set_zlabel('Age', fontweight='bold')

colorPatch = mpatches.Patch(color='#ffc77f', label='Diabetes presence')

ax.legend(handles=[colorPatch])

plt.show()
```

**FIGURE C.1.3 – Weight Distribution Visualization**

“Figure C.1.3” offers a visualization of the weight distribution within the dataset, relative to all three factors. The Python code below outlines the steps for generating this insightful figure.

“Figure C.1.1” demonstrates the correlation between two of the three independent variables to the outcome. The outcome of the heatmap seems rather disappointing at first as it does not showcase an immediate correlation between the data fields and Diabetes presence. However, the absence of 2-dimensional visual proof of linear data does not declare the research indefinite.

“Figure C.1.2” delivers a 3-dimensional scatter plot tying BMI, Diabetes pedigree score, and age to outcomes as related in the data frame. The figure does display a visual to make a case that the BMI, Diabetes pedigree function, and age can correlate to Diabetes, but it is not the most user-friendly visual to interpret, introducing “Figure C.1.3”. “Figure C.1.3” makes use of all three

fields and the outcome, in a more visually appealing case. Upon seeing “Figure C.1.3”, the further away an individual's metrics move from the cluster of black dots representing no Diabetes presence, the more likely the machine is to detect the presence of Diabetes when user input is compared against the practice’s dataset.

Code:

```
X = dataframe.values[:, 0:2]
y = dataframe.values[:, 3].tolist()
y = list(map(int, y))

sample_weight_constant = np.ones(len(X))

descriptiveNoWeight = svm.SVC()
descriptiveNoWeight.fit(X, y)

fig, axes = plt.subplots(1, 1, figsize=(12, 6))

xx, yy = np.meshgrid(np.linspace(0, 85), np.linspace(0, 3))

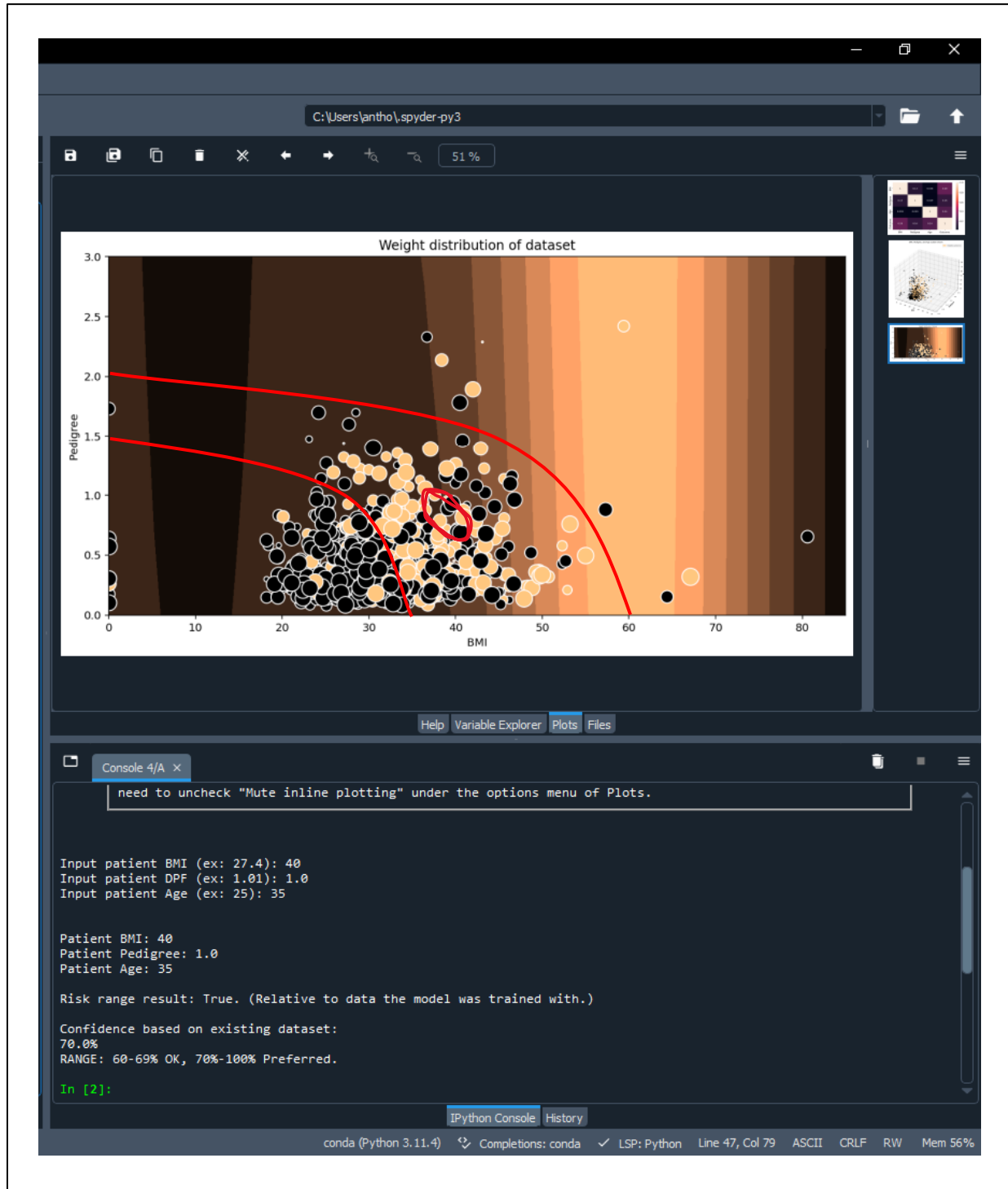
Z = descriptiveNoWeight.decision_function(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, alpha=1, cmap=plt.cm.copper)
```

```
plt.scatter(  
    X[:, 0],  
    X[:, 1],  
    c=varOutcome,  
    s=ageSpace * 3,  
    alpha=1,  
    cmap=plt.cm.copper,  
    edgecolors='white',  
)  
  
plt.title('Weight Distribution of Dataset')  
plt.xlabel('BMI')  
plt.ylabel('Pedigree')  
  
plt.show()
```

For a practical example of “Figure C.1.3” and the corresponding input used for prediction, please refer to the following section.



**FIGURE C.1.4** – Hypothesis, Model Placement, and Mapping of SVM Model

### Product Accuracy:

As previously mentioned, Diabetes imposes itself across a diverse spectrum of individuals, a characteristic that is observable in each of the three preceding figures. Achieving an accuracy rate of approximately 70% to 80% in predicting the presence of Diabetes based on three independent variables represents a noteworthy accomplishment in the context of initial diagnosis. Let it be known that the model lacks a comprehensive biological understanding of the condition. Nevertheless, the model's current accuracy range serves as a valuable tool for preliminary detection.

It is worth noting that in business applications of machine learning solutions, a target accuracy threshold of 90% is desired. However, surpassing this threshold would entail a transition from preliminary Diabetes detection to a more detailed machine-learning solution based on an array of biological tests, including A1c, glucose, ketone, and C-peptide levels. Such an approach, while more accurate, would significantly increase the cost and complexity of the solution's production.

This contextual understanding underscores the importance of achieving a balance between accuracy and practicality when developing machine learning tools for healthcare, particularly in the initial stages of disease detection and diagnosis.

## The Development Process:

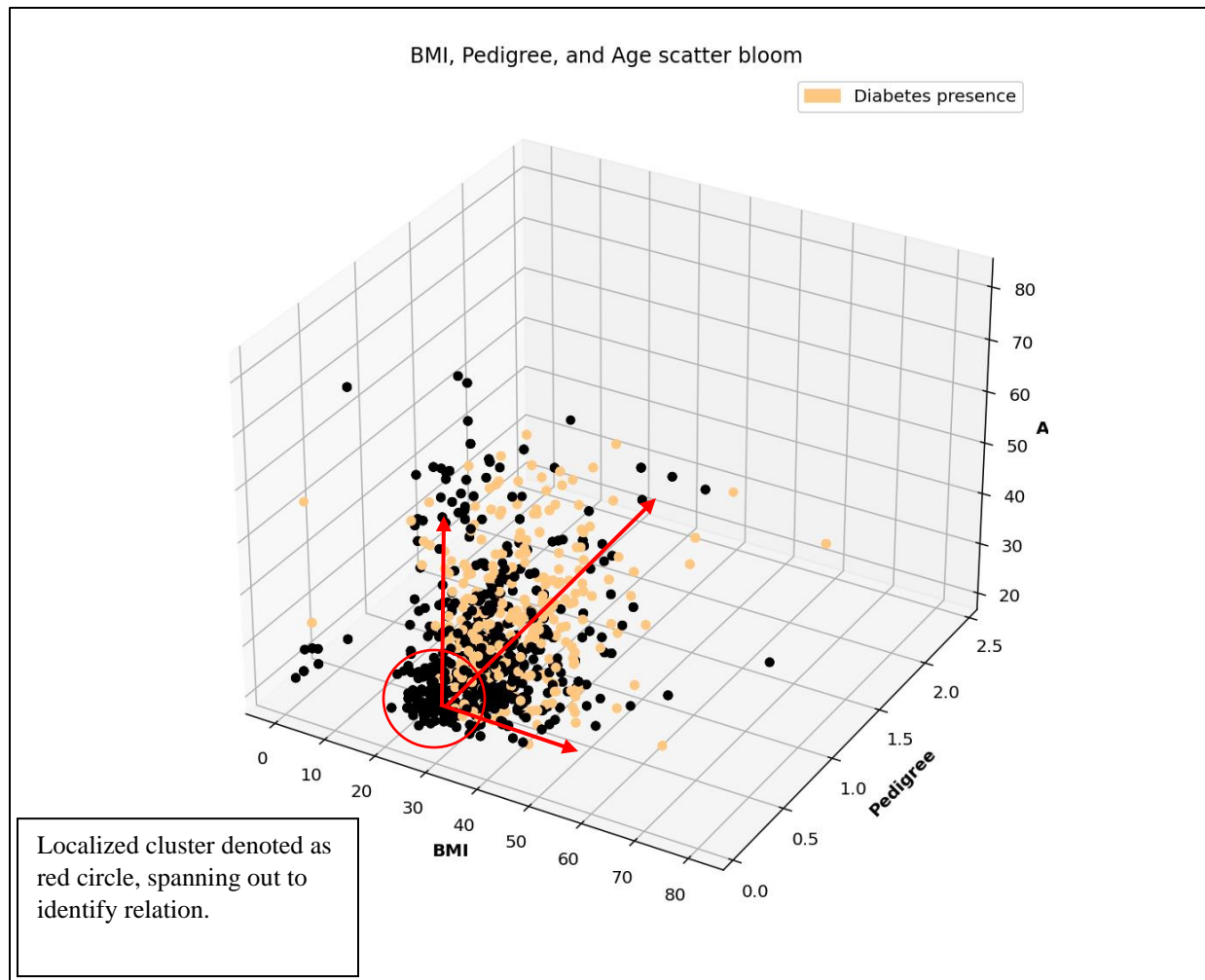
The development of the data product evolved incrementally, employing various plans to demonstrate functionality. The initial visualization (Figure C.1.1) revealed that there exists no immediate linear correlation exceeding thirty percent between any two data fields and the presence of Diabetes. Instead, the most substantial correlation with Diabetes presence was observed when directly comparing each data field with the outcome variable. For instance, the correlation between BMI and the outcome stood at approximately 30%, while the age-outcome correlation neared 25%, and the pedigree-outcome correlation exceeded 15%. It is noteworthy that direct correlations between pairs of data fields and the outcome were not evident, yet the correlation matrix offered limited evidence of relationships among the three measured fields and their connection to Diabetes presence, setting the stage for “Figure C.1.2.”

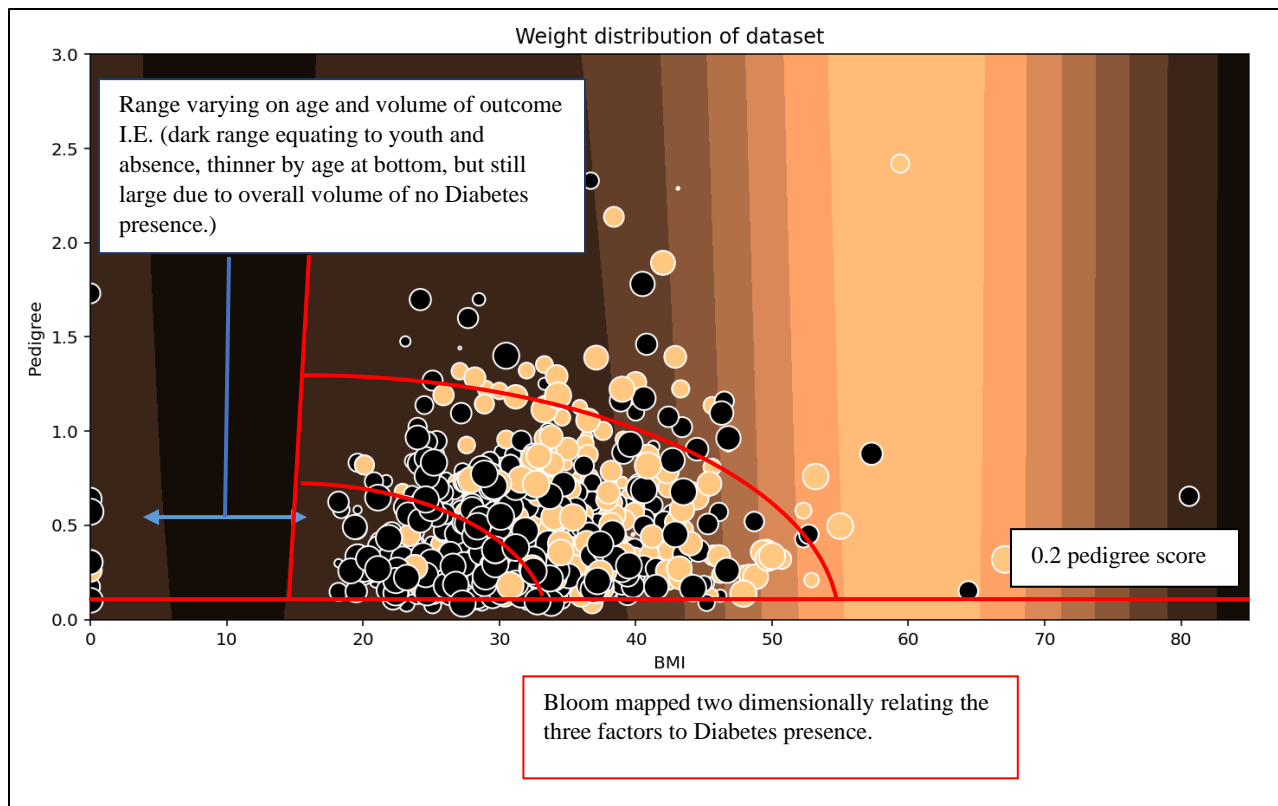
“Figure C.1.2” introduced a three-dimensional scatterplot to assess potential separations between Diabetes presence (represented as 0 for non-existent and 1 for Diabetes presence) and the three data fields: BMI, Diabetes pedigree score, and age. The figure was further developed to integrate light coloring for Diabetes presence and black for its absence, aiding in visual differentiation. The primary purpose of “Figure C.1.2” was to refine and enhance the insights gained from “Figure C.1.1.” The figure vividly illustrated how instances of Diabetes presence extend beyond a localized cluster, distinct from non-existent cases, within the dataset. Additional refinements and optimizations are documented in “Figure D.1.1”, where the three-dimensional scatterplot is revisited and data adjustments are made in comparison to the correlation matrix.

Finally, “Figure C.1.3” represented a further optimization of the data product, building upon the analysis conducted in “Figure C.1.1” and “Figure C.1.2.” This visualization discerned shifts in Diabetes presence influenced by all three factors. Extensive testing, including iterations

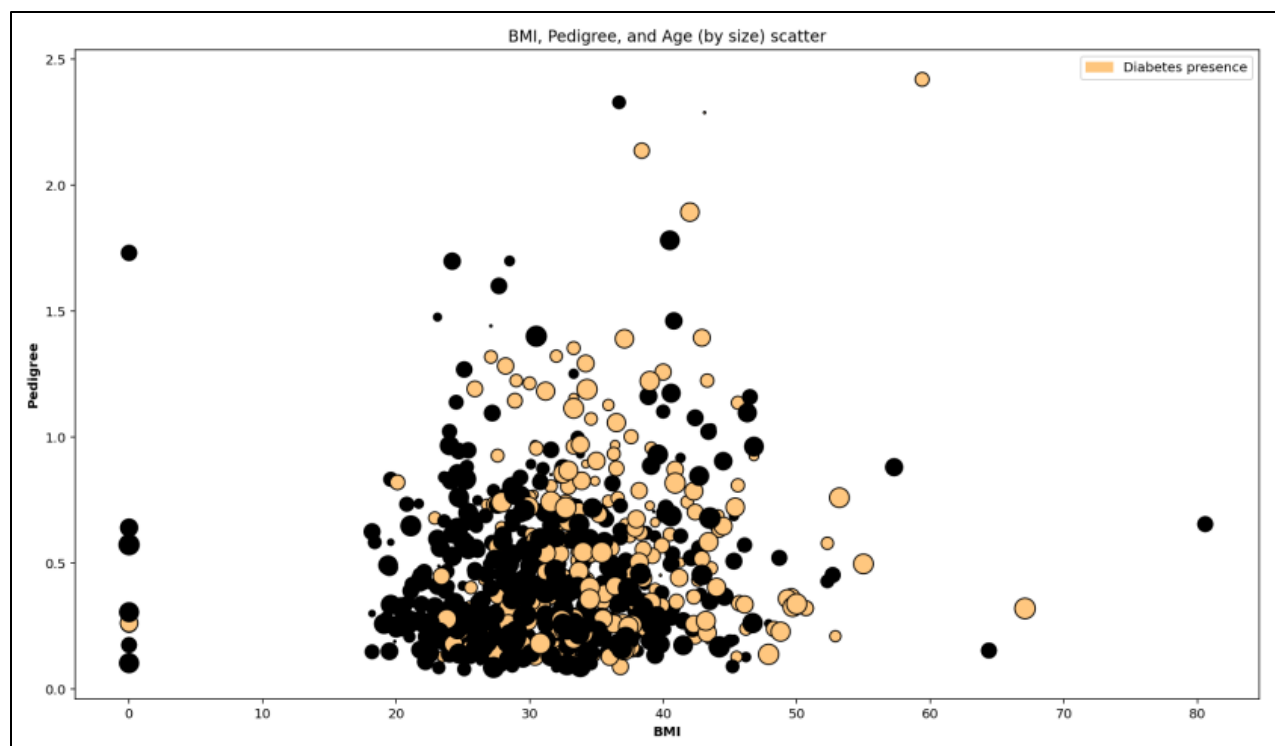
of an unbalanced scale plot, led to the conclusion that the prior two coded visualizations were essential for comparison with the final figure. To explore alterations across various datasets, refer to the revised “Figure *D.1.2*” and “Figure *D.1.3*”.

**FIGURE D.1.1** – Visual difference of all variables to Diabetes presence, different from heatmap.



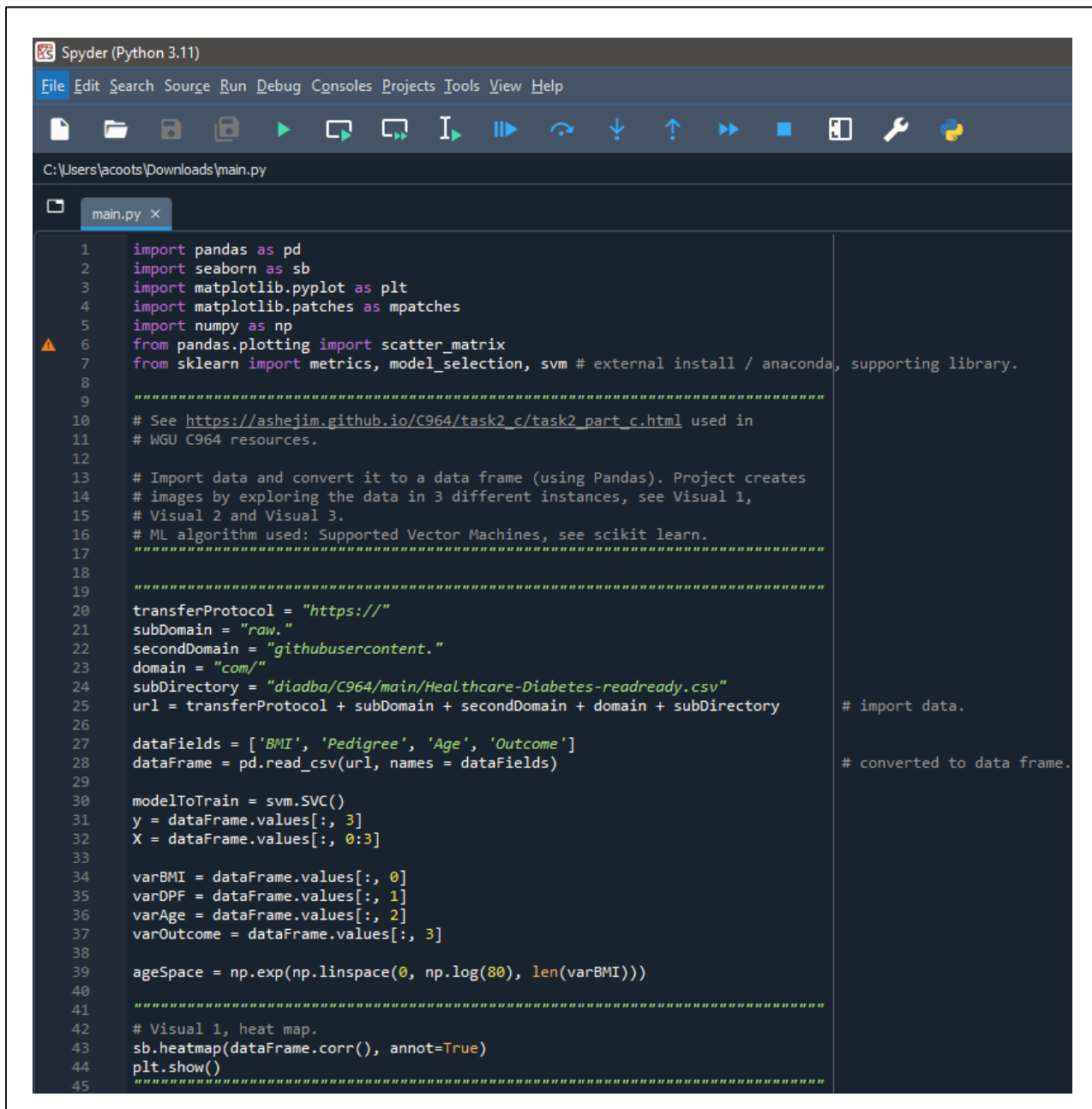
**FIGURE D.1.2** – Three-dimensional graph mapped to scale based on bloom weight.

The data product uses a trained supported vector machine model to make predictions based on input data. Consequently, the plan was devised to present this graphical representation when a user interacts with the console to request a prediction. The underlying design principle aims to provide a visual demonstration of the dataset, allowing them to position their data point along a scale. Refer to “Figure C.1.4” for an illustrative example of this functionality. This particular graph underwent substantial revision, transitioning from an initial two-dimensional scatter plot that proved inadequate in effectively depicting the transition from a cluster denoting a 'non-existent' presence. For reference, please consult the preceding figure showcasing the scatter plot before the core implementation of the balanced scale.

**FIGURE D.1.3** – Figure D.1.2 before revision.

The following figure is the source code (main.py) that is used for the data product. No executables are generated as Spyder provides a user-friendly interface to interact with as part of the data product.

**FIGURE E.1.1**



```

Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\jcoots\Downloads\main.py

main.py x
1  import pandas as pd
2  import seaborn as sb
3  import matplotlib.pyplot as plt
4  import matplotlib.patches as mpatches
5  import numpy as np
6  from pandas.plotting import scatter_matrix
7  from sklearn import metrics, model_selection, svm # external install / anaconda, supporting library.
8
9
10 # See https://ashejim.github.io/C964/task2_c/task2_part_c.html used in
11 # WGU C964 resources.
12
13 # Import data and convert it to a data frame (using Pandas). Project creates
14 # images by exploring the data in 3 different instances, see Visual 1,
15 # Visual 2 and Visual 3.
16 # ML algorithm used: Supported Vector Machines, see scikit learn.
17
18
19
20 transferProtocol = "https://"
21 subDomain = "raw."
22 secondDomain = "githubusercontent."
23 domain = "com/"
24 subDirectory = "diadba/C964/main/Healthcare-Diabetes-readready.csv"
25 url = transferProtocol + subDomain + secondDomain + domain + subDirectory # import data.
26
27 dataFields = ['BMI', 'Pedigree', 'Age', 'Outcome']
28 dataframe = pd.read_csv(url, names = dataFields) # converted to data frame.
29
30 modelToTrain = svm.SVC()
31 y = dataframe.values[:, 3]
32 X = dataframe.values[:, 0:3]
33
34 varBMI = dataframe.values[:, 0]
35 varDPF = dataframe.values[:, 1]
36 varAge = dataframe.values[:, 2]
37 varOutcome = dataframe.values[:, 3]
38
39 ageSpace = np.exp(np.linspace(0, np.log(80), len(varBMI)))
40
41
42 # Visual 1, heat map.
43 sb.heatmap(dataframe.corr(), annot=True)
44 plt.show()
45

```



FIGURE E.1.2



```

Spyder (Python 3.11)
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\acoots\Downloads\main.py

main.py x
43 sb.heatmap(dataFrame.corr(), annot=True)
44 plt.show()
45
46
47
48 # Visual 2, 3-dimensional scatter plot.
49 # Figure
50 fig = plt.figure(figsize = (16, 9))
51 ax = plt.axes(projection = '3d')
52
53 # Grid
54 ax.grid(b = True, color = 'grey', linestyle = '-.', linewidth = 0.3, alpha = 0.2)
55
56 # Plot
57 plot3d = ax.scatter3D(varBMI, varDPF, varAge, alpha = 1, c = varOutcome, cmap = plt.get_cmap('copper'))
58
59 plt.title('BMI, Pedigree, and Age scatter bloom')
60 ax.set_xlabel('BMI', fontweight = 'bold')
61 ax.set_ylabel('Pedigree', fontweight = 'bold')
62 ax.set_zlabel('Age', fontweight = 'bold')
63
64 colorPatch = mpatches.Patch(color='#ffc77f', label='Diabetes presence')
65 ax.legend(handles=[colorPatch])
66
67 plt.show()
68
69
70
71 # Application to algorithm, training algorithm, model creation.
72 # Inspiration from Dr. Jim Ashe, see link: https://wgu.webex.com/recording-service/sites/wgu/recording/946
73 X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y, test_size = 0.3)
74
75 modelToTrain.fit(X_train, y_train)
76
77 y_pred = modelToTrain.predict(X_test)
78
79 patBMI = input("\n\nInput patient BMI (ex: 27.4): ")
80 patDPF = input("Input patient DPF (ex: 1.01): ")
81 patAge = input("Input patient Age (ex: 25): ")
82
83 # Procedure for user to apply model, use IPython Console in Spyder.
84 print("\n")
85 print("Patient BMI: " + patBMI)
86 print("Patient Pedigree: " + patDPF)
87 print("Patient Age: " + patAge + "\n")
88
89 # Application of the model to single input.
90 varResult = str(modelToTrain.predict([[patBMI, patDPF, patAge]]))
91
92 if(varResult == '[0.]'):
93     varResult = "False."
94 else:
95     varResult = "True."
96
97 print("Risk range result: " + varResult + " (Relative to data the model was trained with.)" + "\n")
98 print("Confidence based on existing dataset: ")
99 print(str(round(float(metrics.accuracy_score(y_test, y_pred)), 2) * 100) + "%")
100 print("RANGE: 60-69% OK, 70%-100% Preferred.")
101

```

FIGURE E.1.3

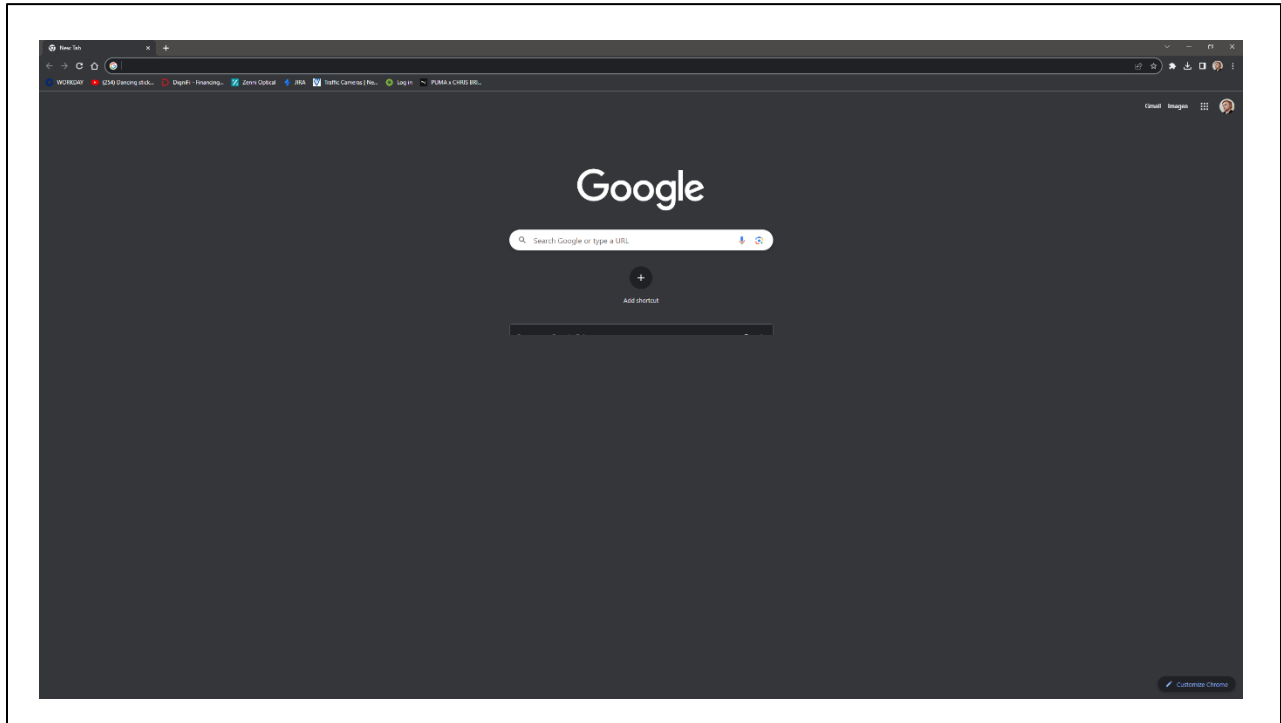
```

102
103
104 # Visual 3, unbalance scatter plot representation.
105
106 X = dataframe.values[:, 0:2]
107 y = dataframe.values[:, 3].tolist()
108 y = list(map(int, y))
109
110 sample_weight_constant = np.ones(len(X))
111
112 descriptiveNoWeight = svm.SVC()
113 descriptiveNoWeight.fit(X, y)
114
115 fig, axes = plt.subplots(1, 1, figsize=(12, 6))
116
117 xx, yy = np.meshgrid(np.linspace(0, 85), np.linspace(0, 3))
118
119 Z = descriptiveNoWeight.decision_function(np.c_[xx.ravel(), yy.ravel()])
120 Z = Z.reshape(xx.shape)
121
122 plt.contourf(xx, yy, Z, alpha=1, cmap=plt.cm.copper)
123 plt.scatter(
124     X[:, 0],
125     X[:, 1],
126     c=varOutcome,
127     s=ageSpace * 3,
128     alpha=1,
129     cmap=plt.cm.copper,
130     edgecolors='white',
131 )
132
133 plt.title('Weight distribution of dataset')
134 plt.xlabel('BMI')
135 plt.ylabel('Pedigree')
136
137 plt.show()
138

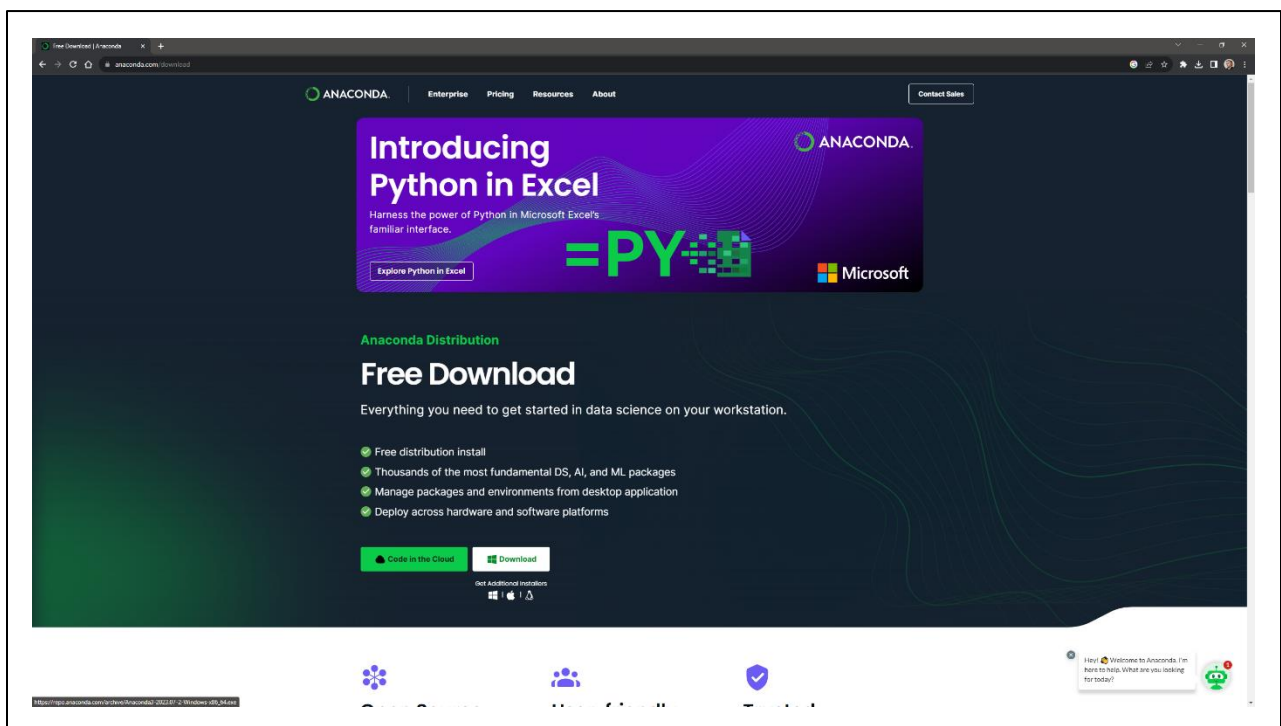
```

Installation Steps:

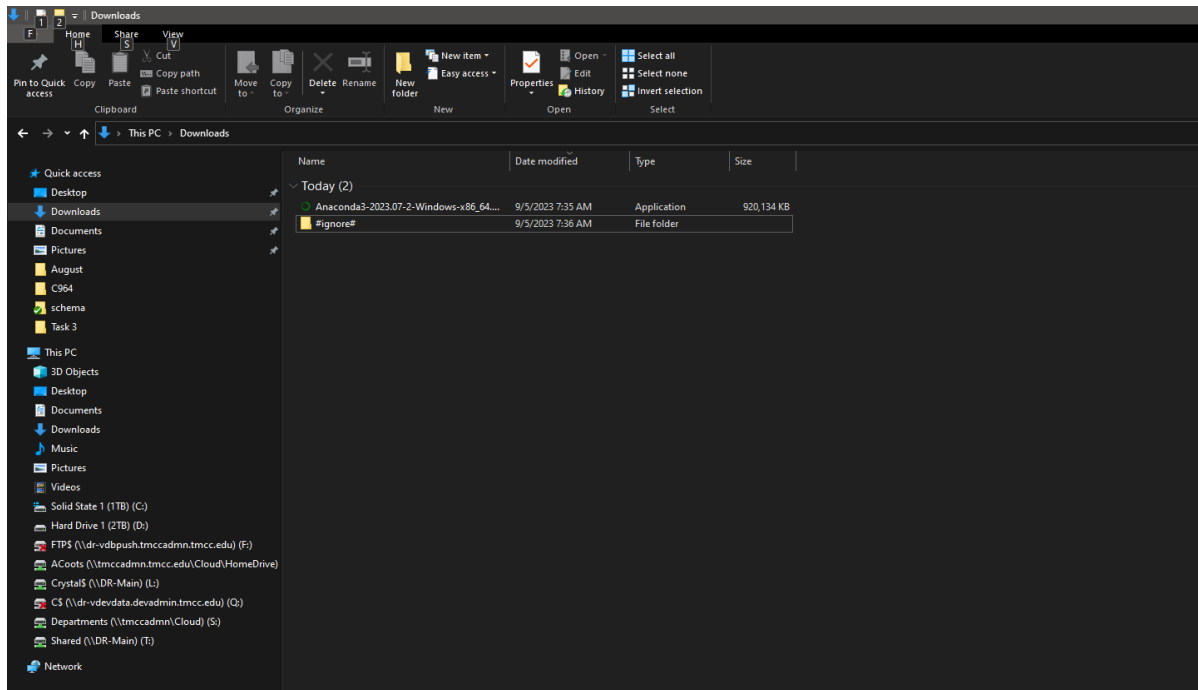
**Step 1: Open a web browser.**



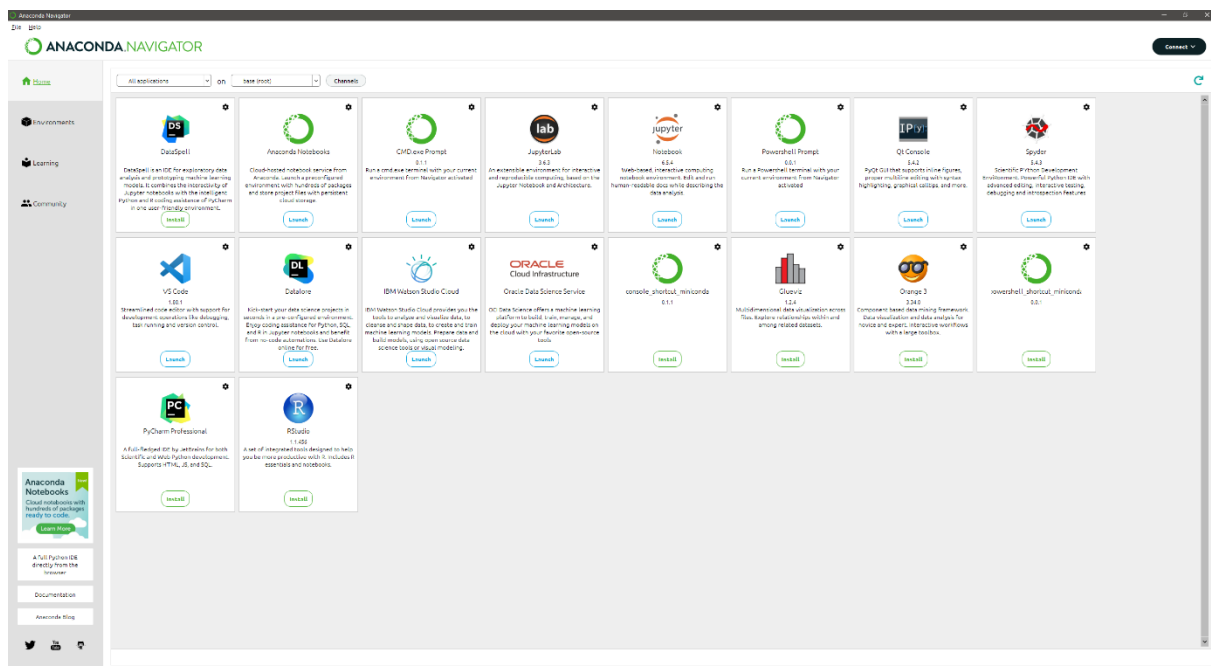
**Step 2: Navigate to <https://www.anaconda.com/download> and select ‘Download.’**



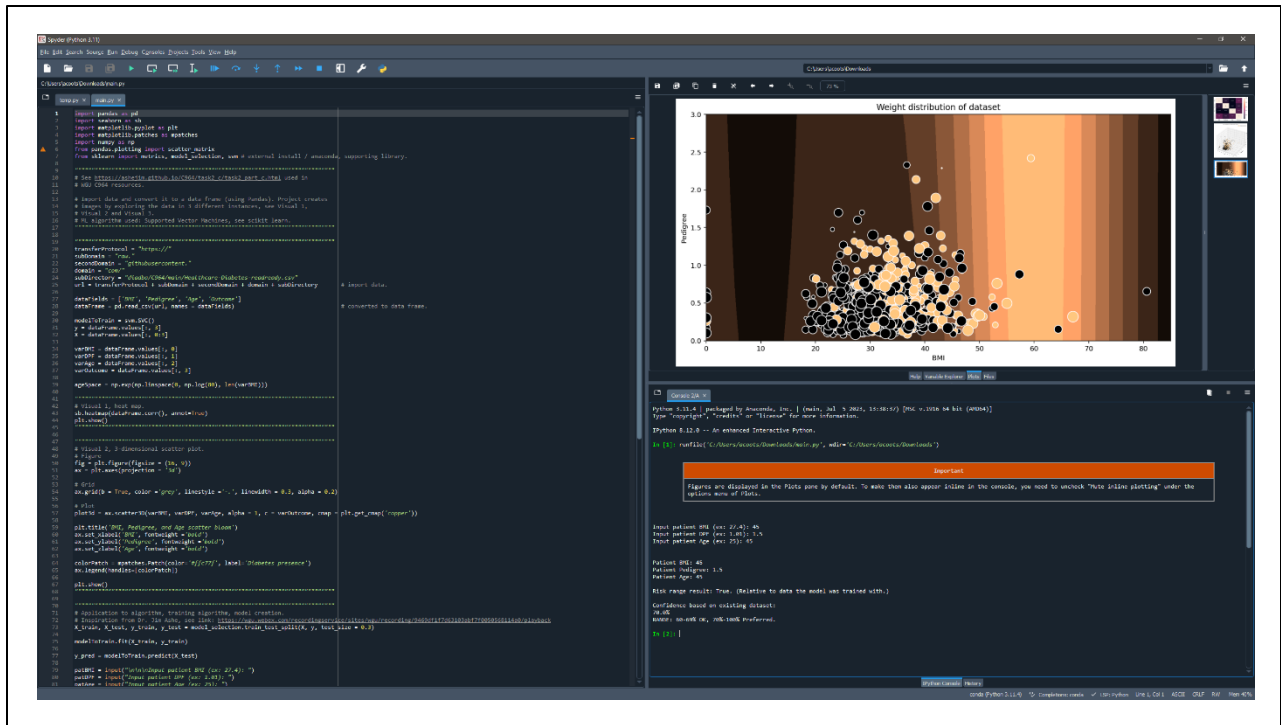
### Step 3: Navigate to download path and run application install.



### Step 4: Open the installation 'Anaconda Navigator.' Select 'Spyder.'



## Step 5: Select File > Open > ‘main.py’ (downloaded with project.)



## References

- Alanazi, A. (2022, March 21). Using machine learning for healthcare challenges and opportunities. <https://www.sciencedirect.com/science/article/pii/S2352914822000739>
- Ashe, J. (n.d.). *Welcome to C964!*. Welcome to C964! - Computer Science Capstone. <https://ashejim.github.io/C964/intro.html>
- Meet virtually with Cisco Webex. anytime, anywhere, on any device.* Cisco Webex Site. (n.d.). <https://wgu.webex.com/recordingservice/sites/wgu/recording/9469df1f7d63103abf7f0050568114a0/playback>
- Pore, N. (2023, August 23). *Healthcare diabetes dataset*. Kaggle. <https://www.kaggle.com/datasets/nanditapore/healthcare-diabetes>