**Computer Science Capstone – C964**

Diabetes Prediction System with Supervised Machine Learning

**Author:** Sochandaling Teng

**Student ID:** 010309126

**Date:** January 26, 2024

Table of Contents

[Part A: Letter of Transmittal 2](#_Toc159486053)

[Part B: Project Proposal Plan 4](#_Toc159486054)

[Project Summary 4](#_Toc159486055)

[Data Summary 4](#_Toc159486056)

[Implementation 6](#_Toc159486057)

[Timeline 7](#_Toc159486058)

[Evaluation Plan 8](#_Toc159486059)

[Resources and Costs 9](#_Toc159486060)

[Part C: Application 11](#_Toc159486061)

[Part D: Post-implementation Report 12](#_Toc159486062)

[Solution Summary 12](#_Toc159486063)

[Data Summary 12](#_Toc159486064)

[Machine Learning 17](#_Toc159486065)

[Validation 18](#_Toc159486066)

[Visualizations 22](#_Toc159486067)

[User Guide 26](#_Toc159486068)

[References 30](#_Toc159486069)

# Part A: Letter of Transmittal

February 2, 2024

Dr. Timothy K. Kim

Director, Luna Medical Center

11911 Artesia Blvd

Cerritos, CA 90701

Dear Dr. Kim,

My name is Sochandaling Teng and I work as a Junior AI/ML Engineer at DataFusion Solutions, one of the top companies in the AI and Machine Learning industry. I am writing to you to present my proposal for implementing a machine learning model within your company. First of all, I will begin with the challenges that most healthcare providers are facing nowadays in the industry. Then, I will demonstrate how this machine learning model will help benefit the organization and improve the healthcare system in the United States with the advancement of new technology.

As you are probably aware, diabetes has become a serious health concern for millions of individuals in the United States. The number of diagnosed diabetes has significantly increased in recent years. According to the Centers for Disease Control and Prevention [CDC] (n.d.), approximately 11.6% of the US population has diabetes and one out of five people has diabetes but has not been diagnosed. This problem has a huge impact on the well-being of individuals and could lead to a substantial economic burden on the nation as it might impose excess costs associated with diabetes.

In responding to this problem, hospitals play a crucial role in managing diabetes and contributing to the economic well-being of the country. It can be done through early detection and diagnosis to provide timely treatment services to individuals with diabetes.

Leveraging AI in the healthcare system, healthcare providers are able to accurately diagnose diseases, provide timely treatments and even prevent them altogether by utilizing data-driven predictive models for early detection of potential diseases and prediction of patient outcomes.

What I am proposing is a data-driven product based on machine learning that can be developed and utilized as a predictive tool within your organization. This data product allows doctors to diagnose diabetes in individuals based on several risk factors associated with diabetes such as the number of pregnancies, glucose level, blood pressure, skin thickness, insulin level, body mass index, diabetes pedigree function and age. It is built to predict whether or not an individual has diabetes using their measurements of those risk factors and trained using diabetes data from the dataset.

I believe that this data product will benefit your organization by providing a predictive analysis tool and facilitating early detection and diagnosis of diabetes. The model can enhance better patient care by analyzing the patient data and predicting the outcome of diagnosis for the doctors, so they are able to identify potential problems, enabling proactive intervention and prevention. Furthermore, the model can improve operational efficiency by streamlining the workflows and eliminating unnecessary procedures such as reading medical images or visualizing patient data. Instead of manually interpreting and processing, the doctors can just use the model to visualize and analyze the patient data to predict the outcome. Additionally, the model can significantly reduce the costs of your organization by optimizing resource allocation through the use of the prediction outcome to adjust with medical inventories or staffing levels, ensuring there is no overstaffing or waste of resources. Overall, leveraging the model will provide your organization with a predictive tool to facilitate diabetes detection and diagnosis, enhance high-quality patient care, improve operational efficiency and reduce costs of the company.

This data product is built as a supervised classification model in the Jupyter Notebook using Python programming language. The model is trained with a diabetes dataset from the Pima Indians Diabetes Database using a logistic regression algorithm to accurately predict whether or not an individual has diabetes. The dataset acquired is publicly available online and not sensitive since it is designed for building the machine learning model purpose. The project is estimated to cost a total of $141,500 for the end-to-end model development including the cost of development, implementation, documentation, maintenance and deployment in the first year, and scheduled to be completed in 3 months. The cost breakdown and timeline milestones will also be provided in detail in the attached proposal.

With my education and training, I have a deep understanding of algorithms and applications in the AI and Machine Learning field. Plus, I have been working on real-world AI projects across various industries; thus, this hands-on experience equips skills to find effective solutions in approaching problems to deliver valuable results for organizations. In addition, I also have experience in analyzing large datasets and developing data-driven applications, enabling me to build a predictive machine learning model to predict future outcomes based on data insights. Combining my experience and skills, I am confident that I am able to successfully develop this machine learning model as a predictive tool to add significant value to your organization and meet your business goals.

I’m looking forward to hearing back from you about this proposal. If you have any questions, please feel free to reach out to us.

Sincerely,

Sochandaling Teng

Senior Engineer, DataFusion Solutions

Phone: +1 (555) 123-4567

Email: sochandaling.teng@dfs.com

# Part B: Project Proposal Plan

## Project Summary

Diabetes has become a serious health concern, affecting millions of Americans in the past few years. Even though healthcare providers are proactively taking action, the number of diabetic people keeps increasing. Most diabetic patients do not have symptoms in the early stage which is a significant challenge for the healthcare providers to diagnose and provide timely treatment services. Relying on just laboratory examination and blood sampling, the detection and diagnosis of diabetes might be late; thus, the treatment is often delayed until the patients reach the last stage. Plus, almost all diagnoses are based on doctors’ judgement and conclusions which is quite subjective due to a lack of scientific data and evidence.

As regards the above matter, that is why each healthcare organization needs a tool to facilitate the diagnosis of diseases, helping doctors detect potential diseases, so they can adopt prevention and provide timely treatment services for patients. This proposal addresses the need of the organization to take control over current issues in the healthcare system through the early detection and diagnosis of diabetes using a data-driven predictive analysis tool. With the dataset and machine learning algorithm, the model is trained to provide data analysis and predict outcomes based on patient data, providing a tool for doctors in diagnosing. Therefore, with the predictive tool, diagnosing diabetes does not solely depend on doctors’ judgement and laboratory results anymore; plus, the doctors are able to accurately diagnose diabetes in future patients based on various risk factors associated with diabetes.

The deliverables of this proposal consist of a supervised machine learning model and a predictive analysis tool to predict whether or not an individual has diabetes. The machine learning model is trained with data from the patient’s diagnostic measurements by using a logistic regression algorithm which is useful for a categorical model (diabetic or non-diabetic outcome). The predictive tool allows the users to enter their diagnostic measurements as the input to the model to predict the outcomes, then it displays the output for the users. Last but not least, the dataset that is used for training the model will also be delivered, attached to this proposal. To implement the tool, users might have to follow the user guide on how to install and execute the program which is provided in detail in this proposal.

The machine learning model will provide healthcare providers with a predictive analysis tool to facilitate early detection and diagnosis of diabetes. This allows the organization to enhance high-quality patient care through data analysis and diagnosis, improve operational efficiency by streamlining workflows and eliminating unnecessary procedures, and reduce the costs for the company with resource allocation and optimization.

## Data Summary

In order to utilize the accuracy of results, the model will be trained with a dataset to learn about data patterns and generalize new data to make accurate predictions. The dataset to be used in this proposal is “Pima Indians Diabetes Database” by UCI Machine Learning (2016) which is available via a CC0: Public Domain License and is publicly online at Kaggle with the URL: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database/data>. The dataset is originally obtained from the National Institute of Diabetes and Digestive and Kidney Diseases with a sample of 768 women (500 non-diabetic and 268 diabetic) from 21 years old and above. It is designed to diagnostically predict whether or not an individual has diabetes based on their diagnostic measurements which are the predictor variables in the dataset. Predictor variables of the dataset consist of eight attributes as the following:

* Number of pregnancies
* Plasma glucose concentration at 2 hours in an oral glucose tolerance test
* Diastolic blood pressure (mm Hg)
* Triceps skin fold thickness (mm)
* 2-Hour serum insulin (mu U/ml)
* Body mass index (kg/m^2)
* Diabetes pedigree function
* Age (years)

Using these attributes, the outcome will be predicted as a binary classification of either 0 or 1. The outcome of 0 indicates that the patient does not have diabetes while the outcome of 1 indicates that the patient has diabetes.

The data lifecycle consists of four stages throughout the model development including acquiring data, preprocessing/cleaning data, analyzing data and visualizing data. The dataset will be first downloaded from Kaggle as a CSV file. Then, several data exploration and analysis techniques will be performed to preprocess the raw data to be ready for the model training including understanding data characteristics and relationships, preprocessing data and visualizing data. The understanding of data will be performed to get a quick overview of variables, types, dimensions and distributions of data in the dataset; thus, any data issues can be identified and mitigated prior to data preprocessing. Common data issues include missing value, duplicate value, inconsistent format and outliers which could cause the model to make inaccurate predictions. After learning about data, preprocessing data is a crucial step in developing the model. This step ensures that all data issues are handled and the data is cleaned and ready for the model training to make accurate predictions. Last but not least, using the descriptive method, data will be visualized into charts and plots which helps to understand the underlying structure, trend, pattern and relationship within the data. Several visualization tools such as count plot, scatter plot, pair plot, histogram and correlation matrix will be displayed and analyzed for model training purposes.

The Pima Indians Diabetes Database used in this proposal is a benchmark for research regarding diabetes classification. Pima Indians are the Native Americans living in Mexico and Arizona, USA who were deemed to have a high rate of diabetes (Schulz et al., 2006); thus, this group can be a significant representative of global health for researchers to conduct studies based on their groups. Furthermore, the feature variables of the data are simple and interpretable which do not require any extensive testing to get the outcome as a binary variable, fitting perfectly for the supervised learning to build a classification machine learning model. In addition, the dataset will be split into training and testing sets to develop the model. By separating training and testing data, the model learns about the relationship within the data to generalize new and unseen data in making accurate predictions. Therefore, we can assess how well the model can perform to provide meaningful and reliable results to us in diagnosing diabetes in patients.

Ethical concerns regarding the use of the dataset include informed consent, privacy and confidentiality. Informed consent should be obtained from individuals whose data is included in the dataset, so they are informed about how their information will be used. Thus, maintaining privacy and confidentiality is important since the data is very sensitive. In response to those concerns, strong data protection measures should be implemented to prevent unauthorized use or disclosure of health information. The use and handling of the data should be aligned and comply with data protection laws such as the Health Insurance Portability and Accountability Act (HIPAA) to protect the rights of individuals in the dataset.

## Implementation

The methodology used in this proposal is **CRISP-DM**. The CRISP-DM (Cross-Industry Standard Process for Data Mining) is a data mining methodology that is widely used in analytics and machine learning projects. The methodology provides a flexible and iterative approach for data mining projects from initial business understanding to final deployment, helping the organization effectively manage the project’s complexity and gain insights to make more informed decisions (Chumbar, 2023). The methodology has six sequential phases applying to this project which are discussed below:

* **Business Understanding:** In this phase, we will define the business problems and requirements to determine the project’s objectives and success criteria, and also communicate with relevant stakeholders prior to formulating the project plan. After reviewing the problem and the proposed solution, we also identify libraries and tools to be used in this project.
* **Data Understanding:** In this phase, we will acquire and explore the data to be collected and used in this project. The dataset will be downloaded from Kaggle and populated into the developing environment. Then, the data will be analyzed to assess its quality and characteristics. It also involves visualizing data into charts and plots, making it easier to assess for relevancy and quality of the data.
* **Data Preparation:** This phase involves data cleaning or preprocessing tasks such as handling data issues, addressing data imbalance, scaling data and splitting data into training and testing sets. We use several functions to check for data completeness in handling data issues such as duplicate value, missing value or null value. The data visualization tool is also used to address imbalance and bias in the dataset which could affect the model’s performance. Additionally, we also apply feature scaling to rescale and standardize the data, ensuring all features in the dataset are in similar scales. Then, we split the data into 20% for testing and the remaining for training, enhancing the model’s performance in generalizing new data to make predictions.
* **Modeling:** In thisphase, we select a machine learning algorithm to train our model. The model will be trained using a logistic regression algorithm for classification. The algorithm is useful for building a supervised predictive model with binary outcomes because it assumes a linear relationship between predictor variables and the outcome. Hence, using the library imported, the model is trained with the training set to understand the impact of the predictor variables on the likelihood of the outcome.
* **Evaluation:** After modeling, we assess and evaluate the model’s performance by comparing the results with the objectives and success criteria set initially in the first phase. The evaluation is performed using several metrics such as accuracy, precision, recall and f1-score. The confusion matrix and ROC Curve will also be assessed and analyzed in evaluating the algorithm’s ability to make predictions.
* **Deployment:** This final phase involves system testing, deploying the model into a real environment and monitoring its performance. The model will be deployed with a user interface, allowing the users to work directly with the model to get the results. The model’s performance will be monitored by continuously training and testing with new data to track and maintain its accuracy and reliability in performing tasks and by encouraging feedback from users about the model to enhance future updates and refinements. Last but not least, the documentation of the model including descriptions, inputs, outputs, user guides and its functionality will be provided and communicated to relevant stakeholders after the deployment.

## Timeline

The milestones of the project map directly with the phases of the methodology to implement the proposed solution. The timeline of each milestone including durations, start dates and end dates is listed in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Milestone** | **Duration** | **Start Date** | **End Date** |
| **Business Understanding**  - Define problems, requirements and scopes  - Define objectives and success criteria  - Set up roles and responsibilities | 1 week | 03/01/2024 | 03/07/2024 |
| **Data Understanding**  - Collect and populate data  - Assess data quality and characteristics  - Analyze and visualize data  - Identify data issues | 2 weeks | 03/08/2024 | 03/21/2024 |
| **Data Preparation**  - Mitigate data issues  - Address class imbalance  - Apply feature scaling and feature engineering  - Split data for training and testing | 3 weeks | 03/22/2024 | 04/11/2024 |
| **Modeling**  - Research and compare algorithms  - Select an algorithm for modeling  - Set up development environment and tools  - Train and validate with the dataset | 3 weeks | 04/12/2024 | 05/02/2024 |
| **Evaluation**  - Select appropriate evaluation metrics  - Assess the model’s performance with initial objectives and success criteria  - Fine-tune the model and perform optimization | 2 weeks | 05/03/2024 | 05/16/2024 |
| **Deployment**  - Conduct system testing with users’ inputs  - Deploy the model into a production environment  - Monitor the model’s performance and data distribution  - Periodically re-train the model with new data  - Finish a summary report of the project and technical documentation of the model | 2 weeks | 05/17/2024 | 05/31/2024 –  ongoing |

## Evaluation Plan

The evaluation plan of the project consists of verification methods to be executed for each phase and validation methods to be implemented upon the completion of the project.

During the **business understanding phase**, an initial meeting with all stakeholders will be conducted to ensure that the problems, requirements and project scopes are well-defined and aligned with expectations. The project plan including objectives and success criteria is also reviewed and approved by the management team to serve as a baseline for validating purposes at the end of the project.

At the **data understanding phase**, the reliability and authenticity of the data are verified in order to populate into a development environment. The characteristics and quality of the data are also verified by identifying data issues such as inconsistencies, missing values, duplicate values, null values or outliers and analyzing the data with visualization tools to ensure that the data is fully assessed and suitable for the project.

In the **data preparation phase**, data preprocessing is verified by ensuring its correctness in applying feature scaling and standardizing the data without any biases. Additionally, data splitting is also verified by ensuring the independence of training and testing sets so that no data leakage or overfitting can occur.

In the **modeling phase**, the generalization ability of the model is verified using the hold-out method with a split of 80% for training and 20% for testing. The hold-out method trains the model with the training set and uses the testing set to test how well the model can perform. In addition, the stability and interpretability of the model are verified by understanding the impact of the feature variables and their significance on the likelihood of the outcome.

In the **evaluation phase**, the metrics are verified with experts’ knowledge and external information to ensure their appropriateness and usefulness in assessing the model’s performance. Evaluation tools used in this project include accuracy, precision, recall, f1-score and more. The model is expected to achieve an accuracy of at least 70% which is considered as good accuracy for the industry standards and an area under the ROC curve of 0.7 or above which is an acceptable level for the predictive model. The confusion matrix is also visualized and analyzed to provide deeper insights into the accuracy and errors of the model in making predictions.

In the **deployment phase**, the functionality of the model is verified by conducting system testing to ensure it meets all functional requirements. The model is deployed and verified with the user’s inputs via a user interface to ensure that the model behaves correctly and makes accurate predictions. After deploying into a production environment, the model will be reviewed to spot any deviations of the model and monitored to maintain its accuracy score of above 70% by tracking its performance and continuously re-training the model with new data.

**Upon the** **completion of the project**, the management team will go through a validation process to evaluate the model if it fulfills its goal of predicting diabetes to ensure that it solves the problems and meets the business objectives. Through the validation, user acceptance testing will be conducted with all stakeholders to gather feedback on the usability and functionality of the model. Lastly, the technical documentation of the model will be reviewed and validated with the management team and a summary report of the project will be delivered and communicated with all relevant stakeholders.

## Resources and Costs

The project is estimated with a total cost of $141,500 for the first year of development, implementation and maintenance. The resources and costs of the project break down into three components such as hardware and software costs, labor costs and environment costs which are listed below:

**Hardware and software costs**

|  |  |  |
| --- | --- | --- |
| **Category** | **Description** | **Total** |
| Workstations | Desktops & Laptops (already owned) | $0 |
| Cloud GPUs | **Google Cloud**  - GPU Model: NVIDIA T4  - Region: Los Angeles (us-west2)  - Number of GPUs: 1  - GPU Time: 730 hours  - Cost: $300/month | $3,600 |
| Cloud Storage | **Google Cloud**  - Storage Class: Standard  - Region: Los Angeles (us-west2)  - Total Storage: 5000 GB  - Cost: $100/month | $1,200 |
| OS License | Microsoft Windows 11 Pro | $200 |
| IDEs | Jupyter Notebook (open-source) | $0 |
| ML Frameworks & Libraries | Scikit-learn (open-source) | $0 |
| Tools & Utilities | Matplotlib (open-source)  Seaborn (open-source)  Pandas (open-source) | $0 |
| **Total** | | **$5,000** |

**Labor costs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Rate** | **Hours Worked** | **Total** |
| ML Engineer | $80/hour | 480 | $38,400 |
| Software Engineer | $75/hour | 480 | $36,000 |
| Data Engineer | $40/hour | 300 | $12,000 |
| QA Engineer | $50/hour | 300 | $15,000 |
| Project Manager | $60/hour | 480 | $28,800 |
| **Total** | | | **$130, 200** |

**Environment costs**

|  |  |  |
| --- | --- | --- |
| **Category** | **Description** | **Total** |
| Data | Pima Indians Diabetes Database (public domain) | $0 |
| Hosting | Google Colaboratory (open-source) | $0 |
| Staff Training | Workshops & Training Materials | $1,800 |
| Infrastructure Monitor & Maintenance | **Atera (all-in-one IT management platform)**  - Software Updates & Patches  - Troubleshooting & Bug Fixes  - Backup & Disaster Recovery  - Plan: All-in-one IT Solution Expert  - Cost: $175/month | $2,100 |
| Security Measures | **ESET (cybersecurity solution platform)** - Server Security  - Endpoint Protection  - Advanced Threat Defense  - Full Disk Encryption  - Plan: ESET Protect Complete  - Number of devices: 50  - Cost: $2,400/year | $2,400 |
| **Total** | | **$6,300** |

# Part C: Application

Files to be submitted along with this document are listed as the following:

* **diabetes.csv** – The dataset used in this model development is “Pima Indians Diabetes Database” by UCI Machine Learning which is publicly available online at Kaggle (<https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database/data>).
* **c962task2.ipynb** – The Jupyter notebook file which consists of the following parts:
  + Libraries and the dataset imported
  + Exploratory data analysis
  + Data visualization with count plot, histogram, scatter plot, pair plot and correlation heatmap
  + Model training with logistic regression algorithm
  + Model evaluation with accuracy, precision, recall, f1-score, false positive rate, confusion matrix and ROC curve
  + User interface for the users to test new data for the model’s prediction

To open with Google Colaboratory, you can upload the notebook file to Google Colab or use the URL below for the notebook hosted in my personal Google Drive:

<https://drive.google.com/file/d/1hBiHHUOz5I8xIlitBLZNPqD6T5HWFVPA/view?usp=drive_link>

To execute the application, you can run all cells in the notebook, then you can access the user interface at the bottom of the page.

# Part D: Post-implementation Report

## Solution Summary

Diabetes has significantly affected individuals and the economy of the country, regardless of preventive measures and actions. This may be caused by failure to diagnose the disease and insufficient scientific data and evidence. Failure to diagnose diabetes may cause the treatment services delayed, leading to tragic results and consequences. Additionally, insufficient scientific data and evidence may also lead to false conclusions and judgements, misleading the information for offering treatment services.

The proposed solution consists of data visualizations with analysis such as count plot, histogram, scatter plot, pair plot and correlation heatmap to learn about relationships within the dataset and visualize the performance of the algorithm. The solution also leverages a supervised machine learning model as a tool for doctors to predict whether an individual has diabetes or not based on their diagnostic measurements on several factors.

The machine learning model was trained with the diabetes dataset consisting of patient measurements, allowing the model to learn about those measurements to predict the outcomes. Furthermore, the model was trained using a logistic regression algorithm for classifying between “diabetes” and “no diabetes”. The model provided an analysis tool and indicator for the doctors to detect and diagnose diabetes in individuals. The analysis and prediction were obtained by using the patient’s data and the model. It was done by first training and developing the model, then accepting the user’s inputs of diagnostic measurements through an interactive user interface. Then, the model used those input values and its predictive ability to make predictions and display the outcomes for the user.

The application of the model provides doctors with a predictive tool to facilitate the diagnosis of diabetes. It allows doctors to adopt necessary prevention and provide timely treatment services for patients. Moreover, diagnosing diabetes will not solely depend on laboratory results and doctors’ judgements anymore, since it uses machine learning and scientific data to make assumptions and conclusions, allowing doctors to enhance better patient care. Last but not least, using the model also enables the organization to improve its operational efficiency by streamlining the workflows, eliminating unnecessary procedures and reducing costs with resource optimization.

## Data Summary

The dataset used for this project was the “Pima Indians Diabetes Database”, owned by UCI Machine Learning (2016). It was acquired and downloaded from Kaggle at the URL: <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database/data>. The dataset was extracted from the National Institute of Diabetes and Digestive and Kidney Diseases to serve as a benchmark for machine learning purposes. The format of the dataset is a CSV file consisting of eight diagnostic predictor variables and one target variable with a sample of 768 women as shown in **Figure 1**. Predictor variables include risk factors that are associated with diabetes such as number of pregnancies, blood pressure, age, body mass index and more. With these predictor variables, the target variable can be predicted which is the outcome. The outcome of 0 indicates a negative in diabetes while the outcome of 1 indicates a positive in diabetes.

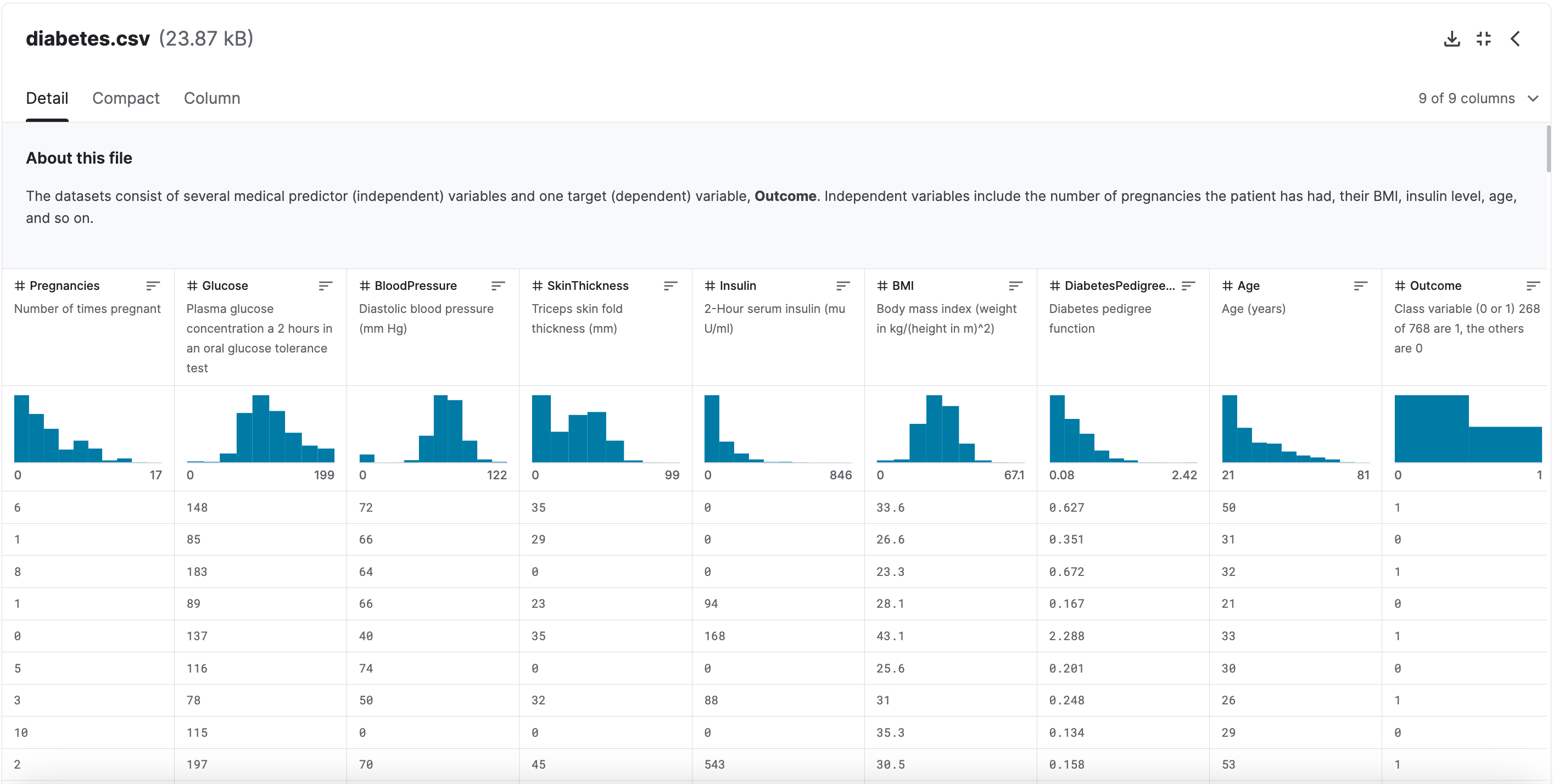
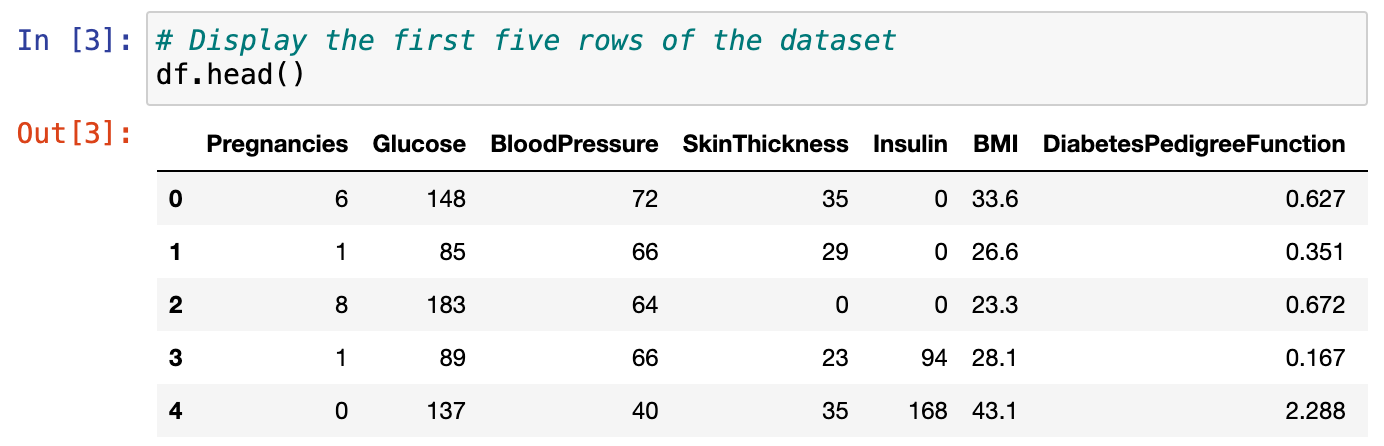


Figure 1. Pima Indians Diabetes Database

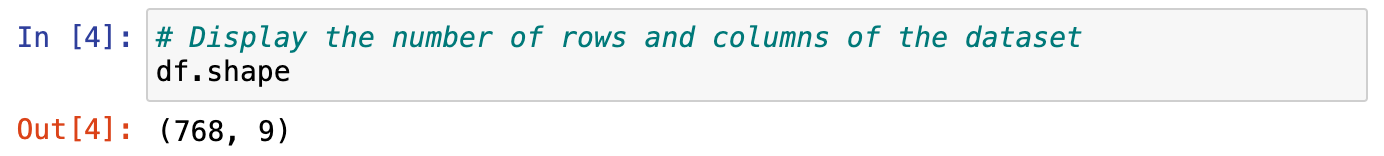
Upon the initial inspection of the dataset, we noticed that all data in each column are numeric, making it suitable and beneficial for machine learning model development. Numeric data types can be easily visualized and analyzed using various libraries such as Pandas, Matplotlib and Seaborn. After being downloaded and populated into a development environment, the dataset was loaded into the data frame by using the Pandas library with the **read\_csv()** function.



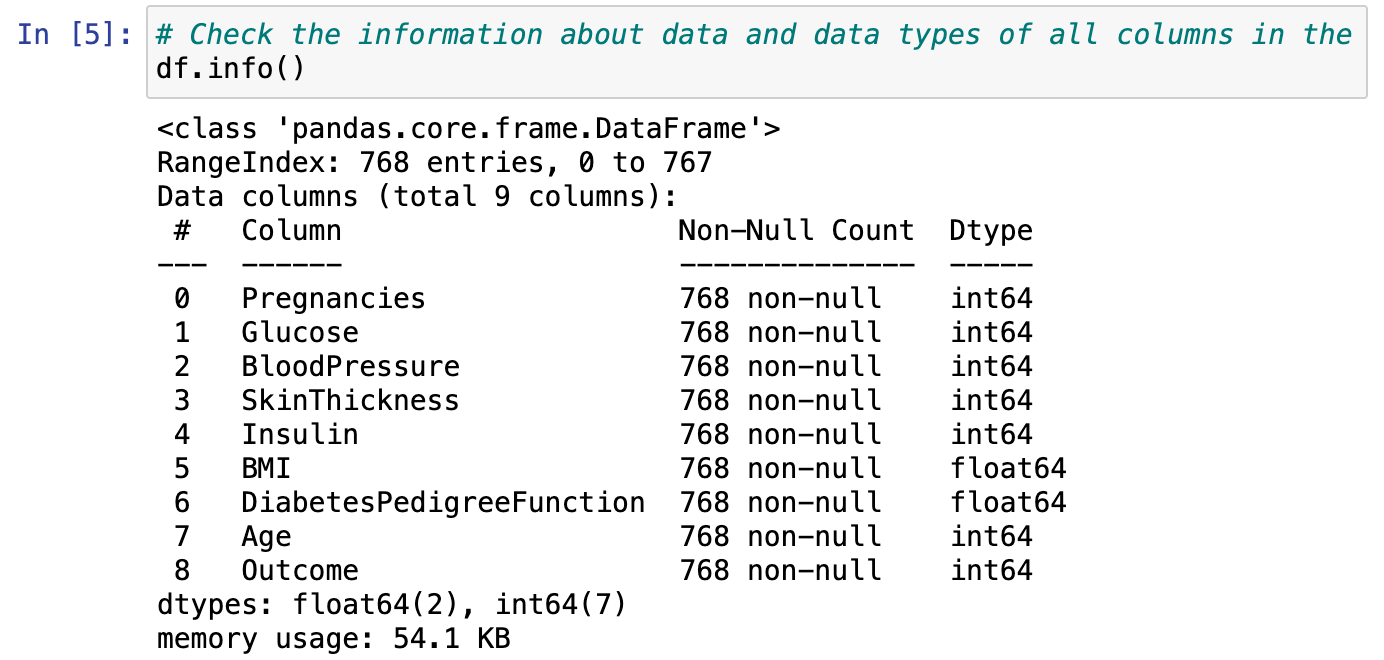
After loading the dataset, the raw data must be processed and ready for modeling. Hence, an exploratory data analysis (EDA) was performed with several techniques including understanding data, preprocessing data and visualizing data. Data understanding was performed to get an initial overview of the structures and characteristics of the data such as variables, types, dimensions and distributions of the data. First, we displayed the first several rows of the dataset using the **head()** function to verify that the dataset was imported correctly.

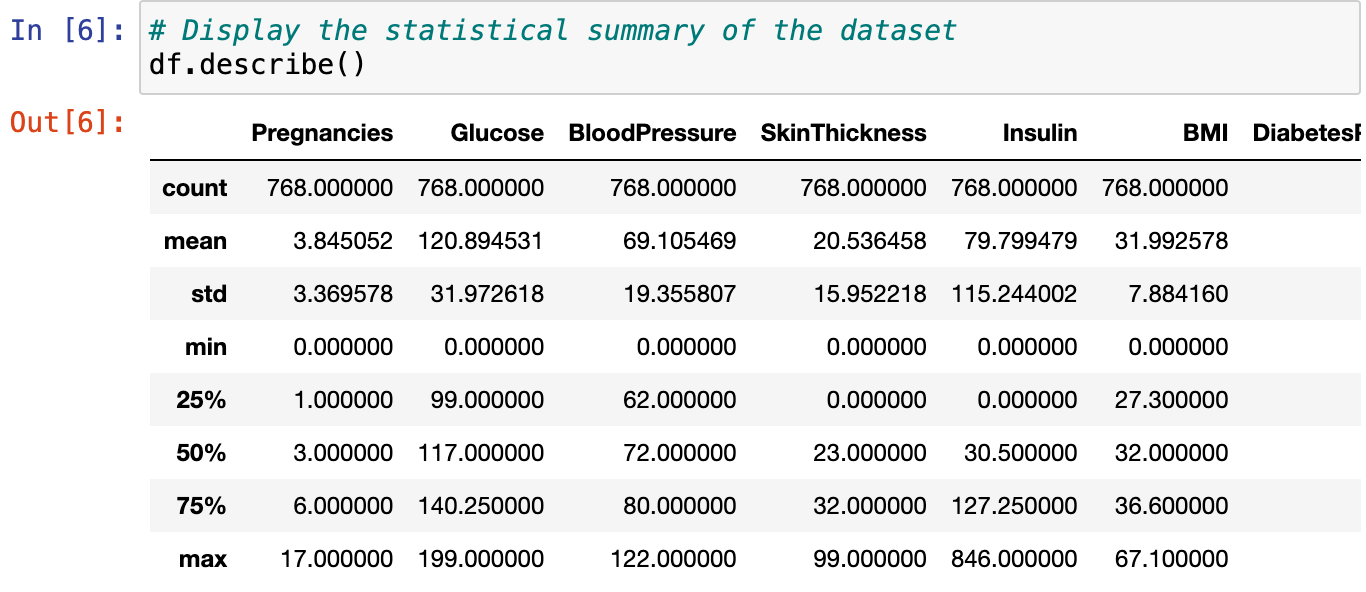


Then we checked the dimensions of the dataset using the **shape** attribute to display the number of rows and columns of the dataset which is useful for later reshaping and scaling.

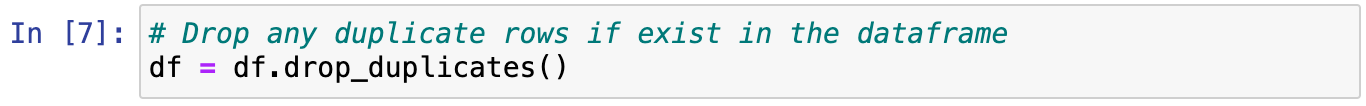


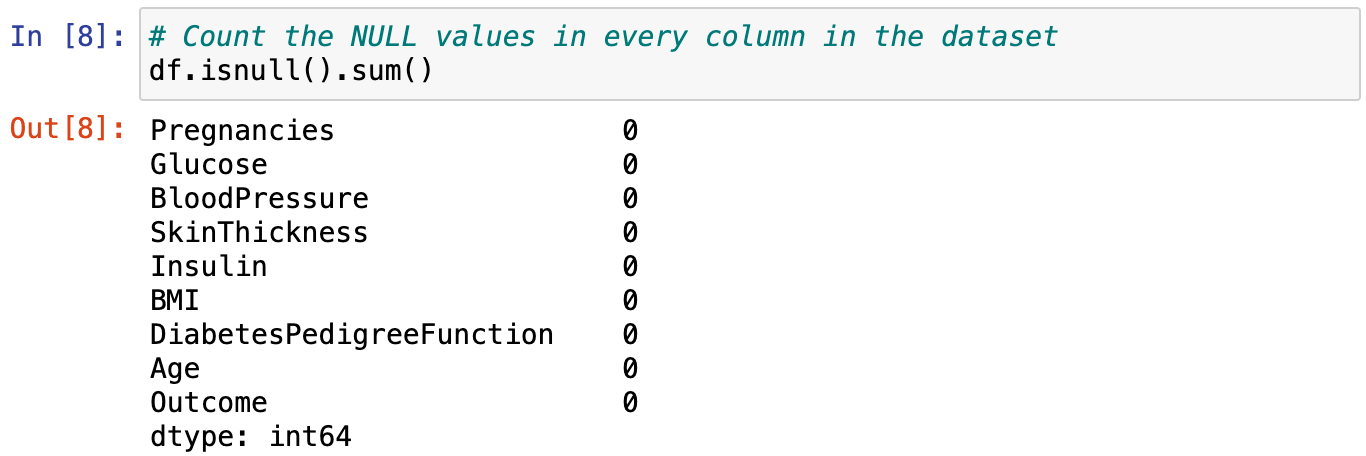
Additionally, the **info()** function was used to retrieve a summary of the dataset including column names, data types and memory usage. A descriptive statistics summary of the dataset was also retrieved by using the **describe()** function to display count, mean, standard deviation, minimum, and maximum, allowing us to assess the quality of the data.



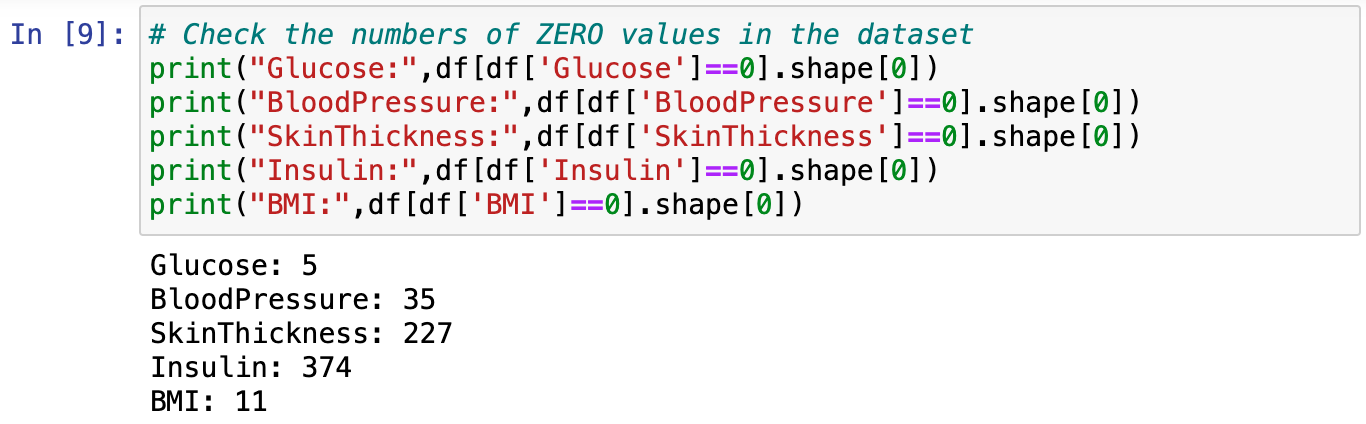
****

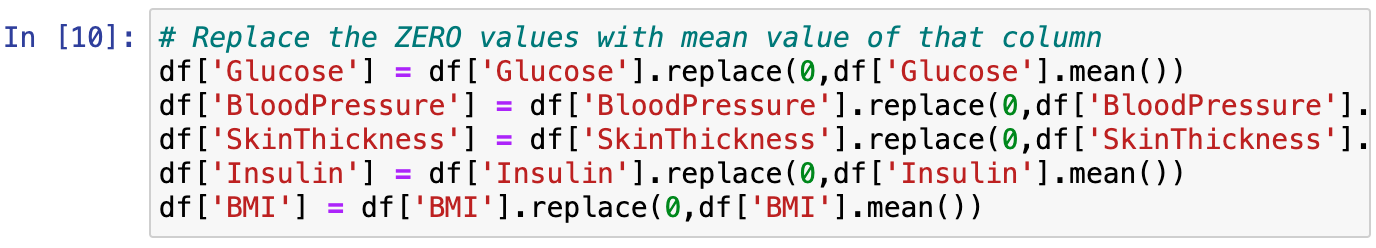
Upon the initial understanding of the dataset, we addressed missing values in the data, allowing us to handle and mitigate those issues in the next step. Then, the data was cleaned to ensure all data issues were eliminated. We used the **drop\_duplicates()** function to drop duplicate rows in the data frame and checked for null values in the dataset with the **isnull().sum()** function.





Based on the descriptive statistical summary during initial data understanding, we noticed that the minimum value of columns “Glucose”, “BloodPressure”, “SkinThickness”, “Insulin” and “BMI” was zero which did not make sense and could be missing data in the dataset. Thus, we checked and replaced those zero values with the mean value for each column in the dataset.





After cleaning the data, we visualized the data into charts and plots using visualization tools from Matplotlib and Seaborn libraries including histogram, count plot, scatter plot, pair plot, and correlation heatmap which will be illustrated later in this proposal. Using the count plot, we checked the class balance of the target variable to see its distribution and select the right approaches to train the model. Based on **Figure 2** shown below, the result found that there was a class imbalance of the target variable which was also known as “Outcome”, meaning that there were more non-diabetes than diabetes in the dataset. This reflects the reality of the healthcare system and indicates a good sign that there are fewer diabetic people; however, using a dataset with the imbalanced class to train the model might lead to bias and poor generalization, making the model less accurate in making predictions.

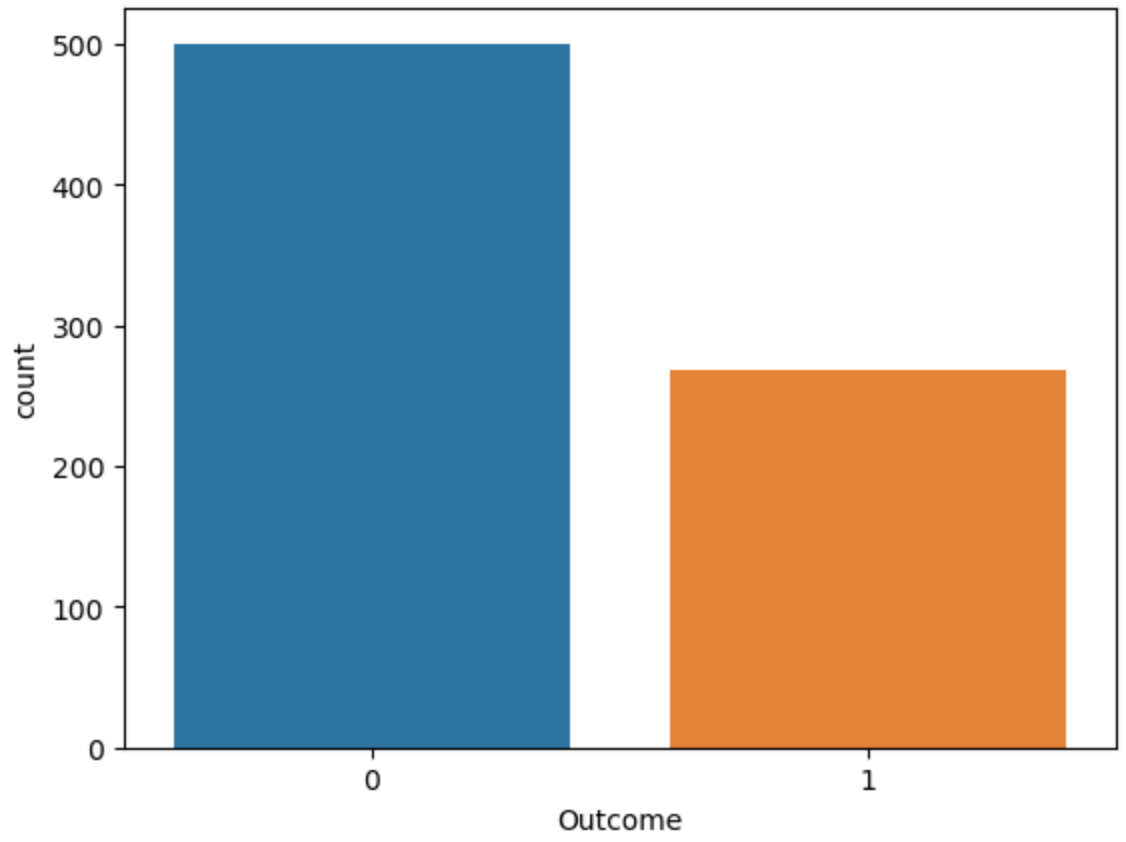
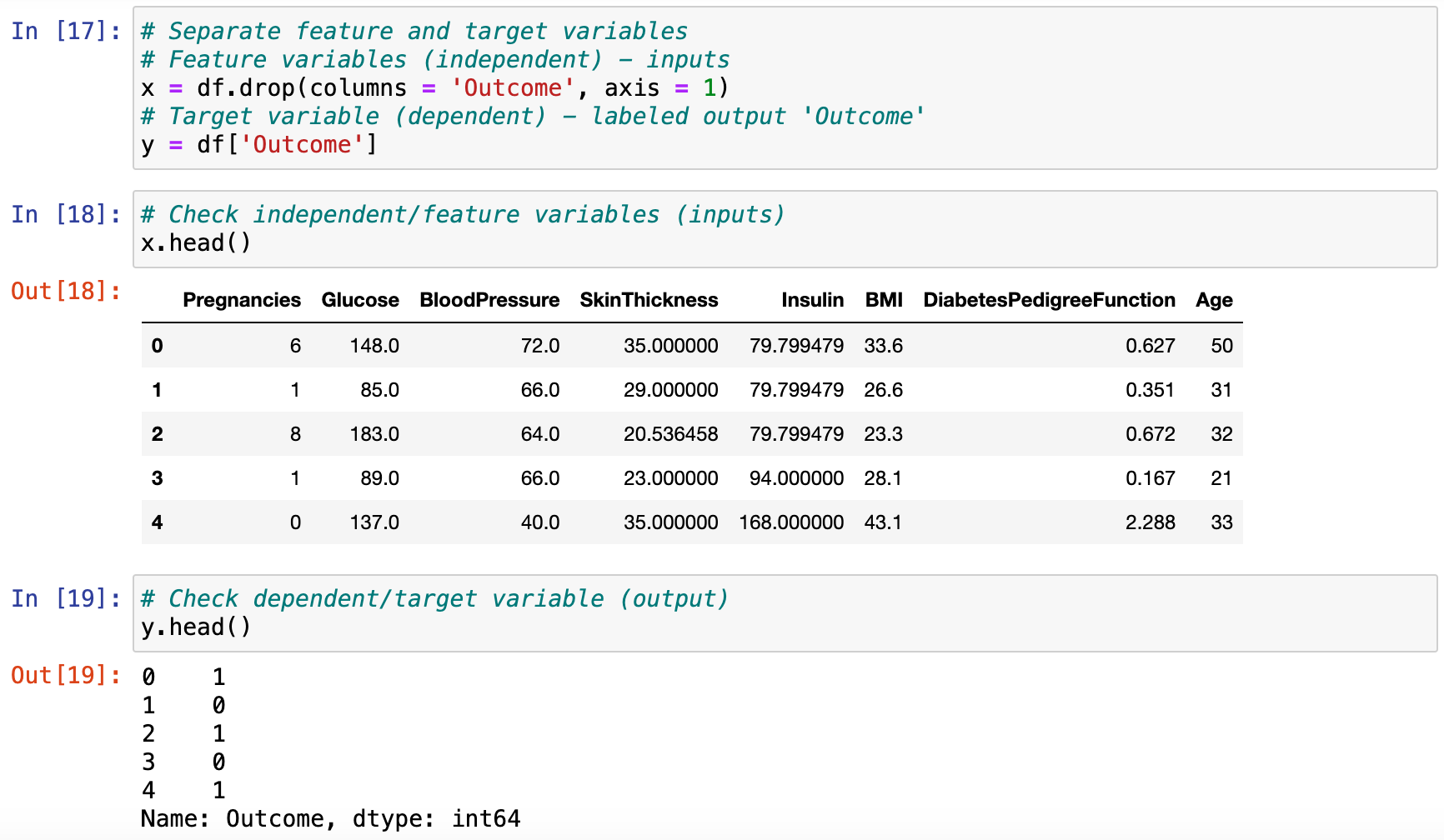
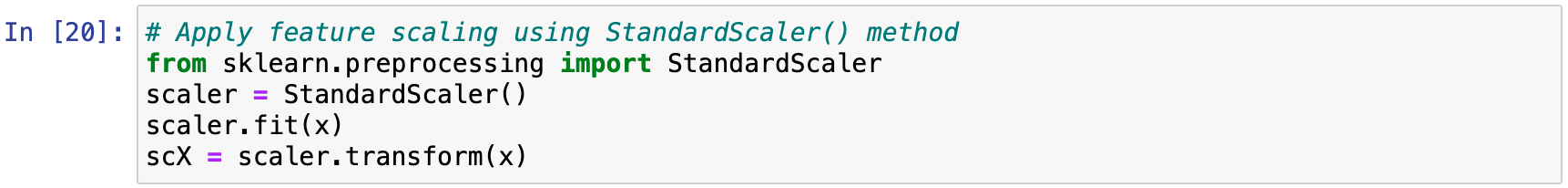


Figure 2. Count plot of each Outcome class

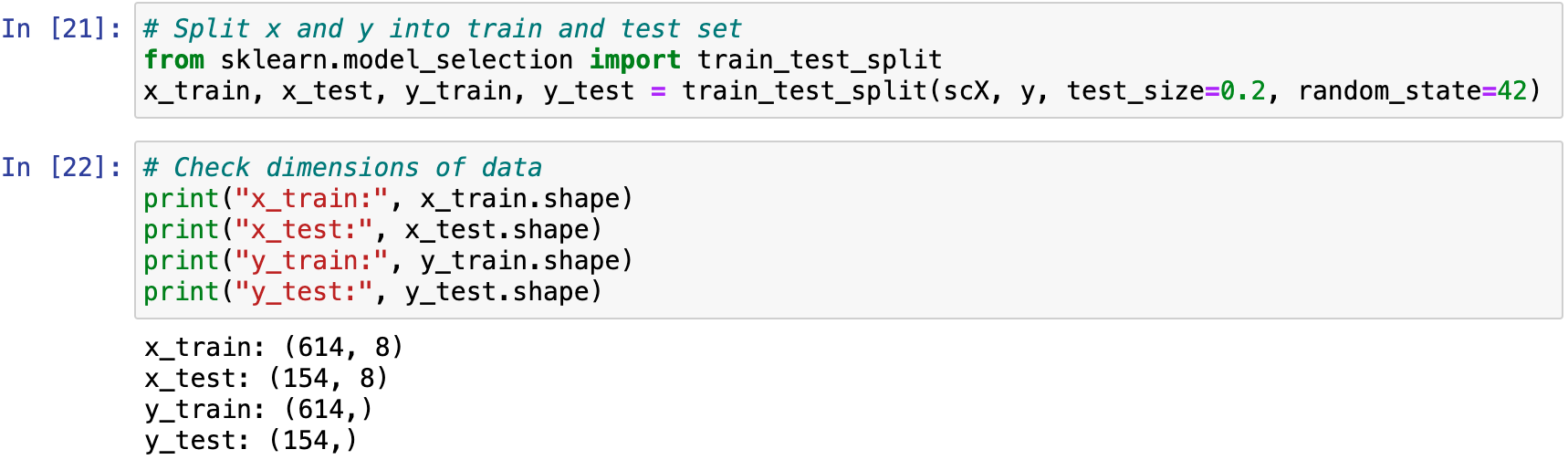
Therefore, to prevent leaking the outcome predictions, we separated the data into feature and target variables to make sure that the model makes predictions from the feature variables only, not from the target variable that it is trying to predict. Target variable leakage might lead to perfect predictions since the model has poor discrimination ability and poor generalization on unseen data.



Furthermore, the values of feature variables were in different ranges and magnitudes, affecting the decision boundary of the model, so we rescaled all features into similar scales to equalize its impact on the target variable by applying feature scaling with the **StandardScaler()** function from Sklearn library.



Then, using the **hold-out method**, we split the dataset randomly into **80% for training** and **20% for testing** by using the **train\_test\_split()** function from the Sklearn library. The training set was used to train the model during the modeling phase and is optional to be used during the evaluation phase for model monitoring and finetuning. However, the testing set was used in the evaluation phase to assess how well the model can perform in generalizing unseen data and making predictions. In addition, during the splitting, we set the **random\_state = 42** to control its consistency in generating the results, so the same training and testing set are produced each time we run the code, avoiding biases in the dataset splitting process.

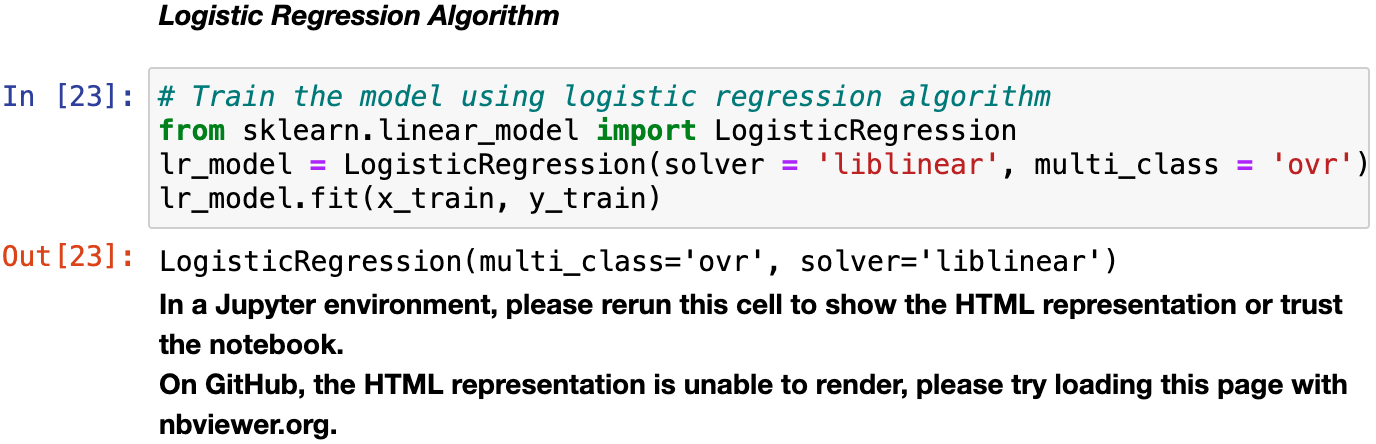


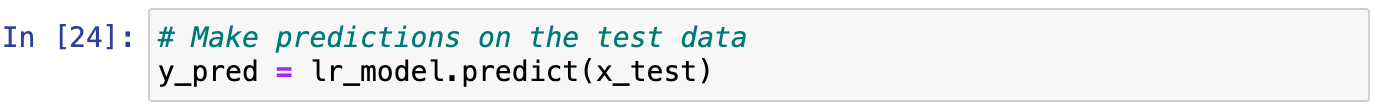
## Machine Learning

The proposed solution is a **supervised machine learning model** which is developed using a **logistic regression algorithm** for the purpose of diagnosing diabetes in individuals. The model predicts the possibility of individuals having diabetes based on their diagnostic measurements, enabling doctors to adopt necessary prevention and provide timely treatment services for patients. With the predictive tool, diabetes is manageable and controllable with the early detection and diagnosis, early intervention and preventive care planning by the healthcare providers.

For the purpose of making predictions, the logistic regression algorithm assumes a linear relationship between the predictor variables and the target variable. The model is trained with preprocessed datasets to learn the relationship between those variables, then the algorithm uses the decision threshold to predict the binary possible outcomes which are “diabetes” or “no diabetes” based on the predictor variables.

The **Scikit-learn library** was imported and used with the **LogisticRegression** model. Since the dataset has a small size and the model aims to predict a binary classification, we used **solver = ‘liblinear’** and **multi\_class = ‘ovr’** for the modeling as shown below. The reason we used liblinear solver is that it can handle sparse data and prevent overfitting, plus it can be extended to handle multi-class classification using one-vs-rest (OvR), leading to a more robust and accurate model. Then, we started training the model with the training set split during the data preparation phase. After training, we used the model to make predictions on the testing set to see how well it can perform in generalizing new data to make predictions. Using the **hold-out method**, the dataset was split to separately train and test the model, providing insights for evaluating the model’s performance and ability to make accurate predictions which will be discussed in the next section.



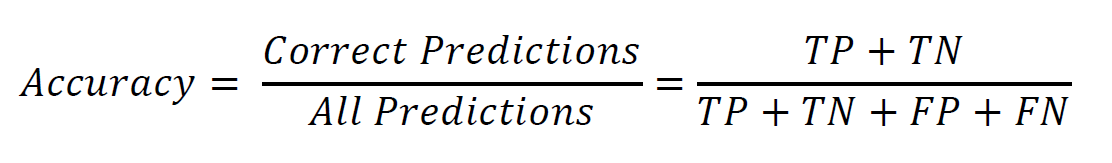


The **LogisticRegression** is a good choice for this project to build an accurate predictive machine learning model. It is designed for binary classification problems where the outcome has two classes, making it well-suited for this proposal in making predictions. Based on the algorithm’s assumption of linear relationship between the predictor variables and the outcome, it provides insights for healthcare providers to easily understand and interpret the results from the model. Furthermore, the algorithm is scalable as it provides various tools during modeling to handle the variation of datasets and the classification tasks such as solver and multi\_class parameters. Plus, it also allows for further fine-tuning by adding more predictor variables or training with larger datasets to improve the model’s performance.

Logistic Regression has been successfully implemented in developing several prediction models to be used in healthcare systems, especially to assist in clinical decisions (Shipe et al., 2019), although other algorithms such as **Support Vector Machines (SVM)** or **Decision Trees** are widely being used for machine learning. The Decision Tree might be **prone to overfitting** due to its sensitivity to data changes and irrelevant features, unlike the logistic regression, which has feature selection techniques to handle those issues during the data preparation phase. In terms of the algorithm’s interpretability, the SVM might be **more difficult to interpret** than the Logistic Regression since the SVM does not show an individual feature importance contributing to the decision outcome, so the model cannot provide a meaningful result for healthcare providers to use as their reference point in making decisions. Based on the context of this proposal, **Logistic Regression** is the **standout choice** and **well-suited** for training and developing our predictive machine learning model.

## Validation

After making predictions, the model’s performance was assessed using its accuracy in making correct predictions. **Accuracy** is the ratio of correct predictions out of total predictions made. It was assessed using the **hold-out method** with training and testing datasets.



The goal set initially in this proposal was to achieve accuracy above 70% with the testing set. As a result, the LogisticRegression model achieved an **accuracy score of 76.62%**, which was acceptable for the industry standard and above the performance baseline. To further evaluate the performance of the model, a **Confusion Matrix** was visualized and analyzed to assess the detailed breakdown of the model’s predictions, which is shown in **Figure 3** below.

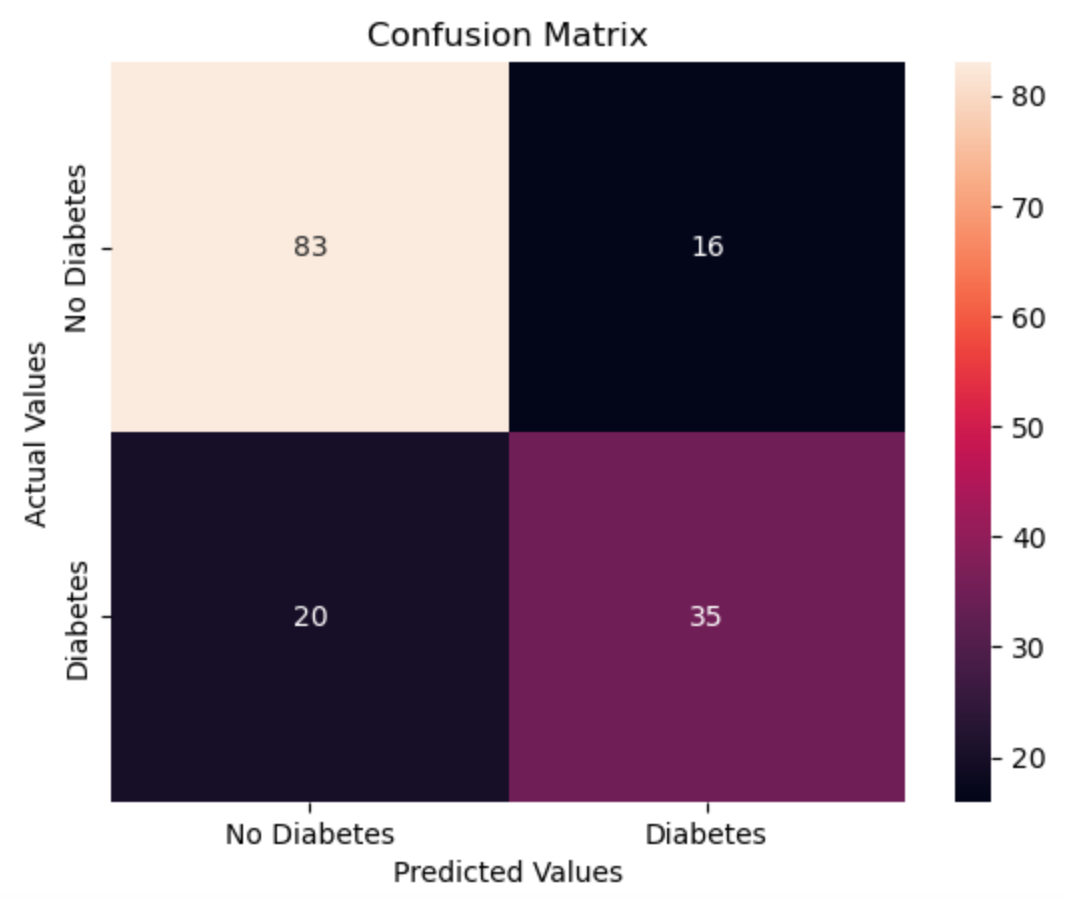


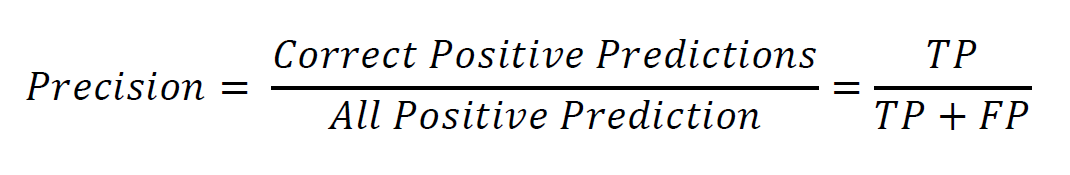
Figure 3. Confusion matrix of Logistic Regression model

The confusion matrix provides the count of four metrics as detailed in the following:

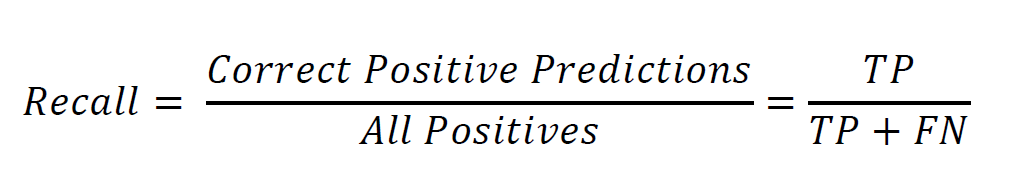
* True Negative (predict: "no diabetes" | actual: "no diabetes") = 83 people
* False Positive (predict: "diabetes" | actual: "no diabetes") = 16 people
* False Negative (predict: " no diabetes" | actual: "diabetes") = 20 people
* True Positive (predict: "diabetes" | actual: "diabetes") = 35 people

However, assessing the accuracy to conclude the model’s performance may not be enough, especially when the dataset has an imbalanced class. Therefore, with the confusion matrix, we computed several more metrics to assess the model’s performance such as precision and recall.

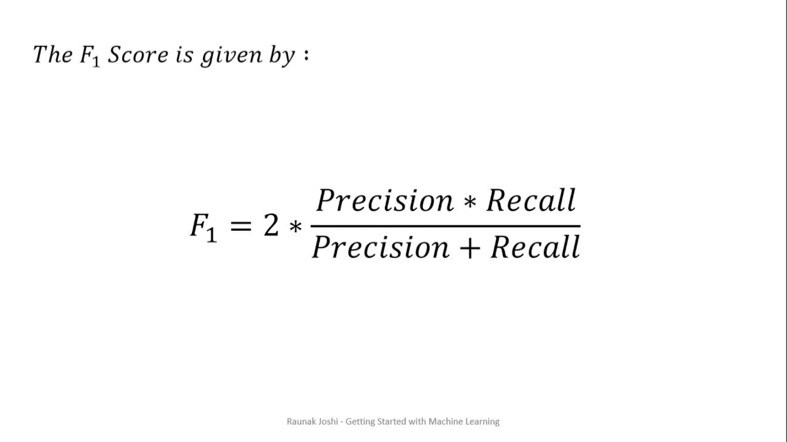
**Precision** is the ratio of correct positive predictions out of all positive predictions, showing how precise the model can predict out of all predicted positives.



**Recall**, also known as **Sensitivity**, is the ratio of correct positive predictions out of all actual positives, showing how well the model can predict actual positives right.



In the context of this proposal, the recall/sensitivity score is an important metric to be assessed to avoid False Negatives. False Negatives occur when the patient is incorrectly predicted as non-diabetic while they are actually diabetic. This could lead to severe implications of failed diagnosis such as delayed treatments and incurred associated costs. Therefore, we aim to maximize the recall score of the model. The model achieved a **precision score of 68.63% and a recall score of 63.63% on the testing set**. However, since the class was imbalanced, the precision and recall scores were further assessed **using the weighted average method**, resulting in the **precision and recall of 76.31% and 76.62% respectively**. Concerning precision and recall, there is a trade-off concept between these two metrics in binary classification problems, meaning that increasing precision might reduce recall, and vice versa. In this case, we can use the **F1-score** to balance between precision and recall.



The goal of the model is to get a high F1-score which indicates both high precision and high recall of the model, minimizing biases towards one metric and balancing the trade-off. As a result, the model achieved the **F1-score of 66.04% on the testing set** and **76.41% using the weighted average method**. Additionally, the ROC-AUC metric was also used to evaluate the model’s discriminative ability to distinguish between classes. The goal is to get a higher AUC score so that the ROC curve leans closer to the top left corner, indicating better performance of the model. As a result, the model achieved the **AUC score of 0.737** which was above the initial baseline of 0.7. As illustrated in **Figure 4** below, the **ROC curve** (red line) leans toward the top left corner, distancing from the diagonal line (blue line) which is a random chance classifier with no discriminative power. This indicates that the model has a moderate discriminative power at classifying instances and making accurate predictions.

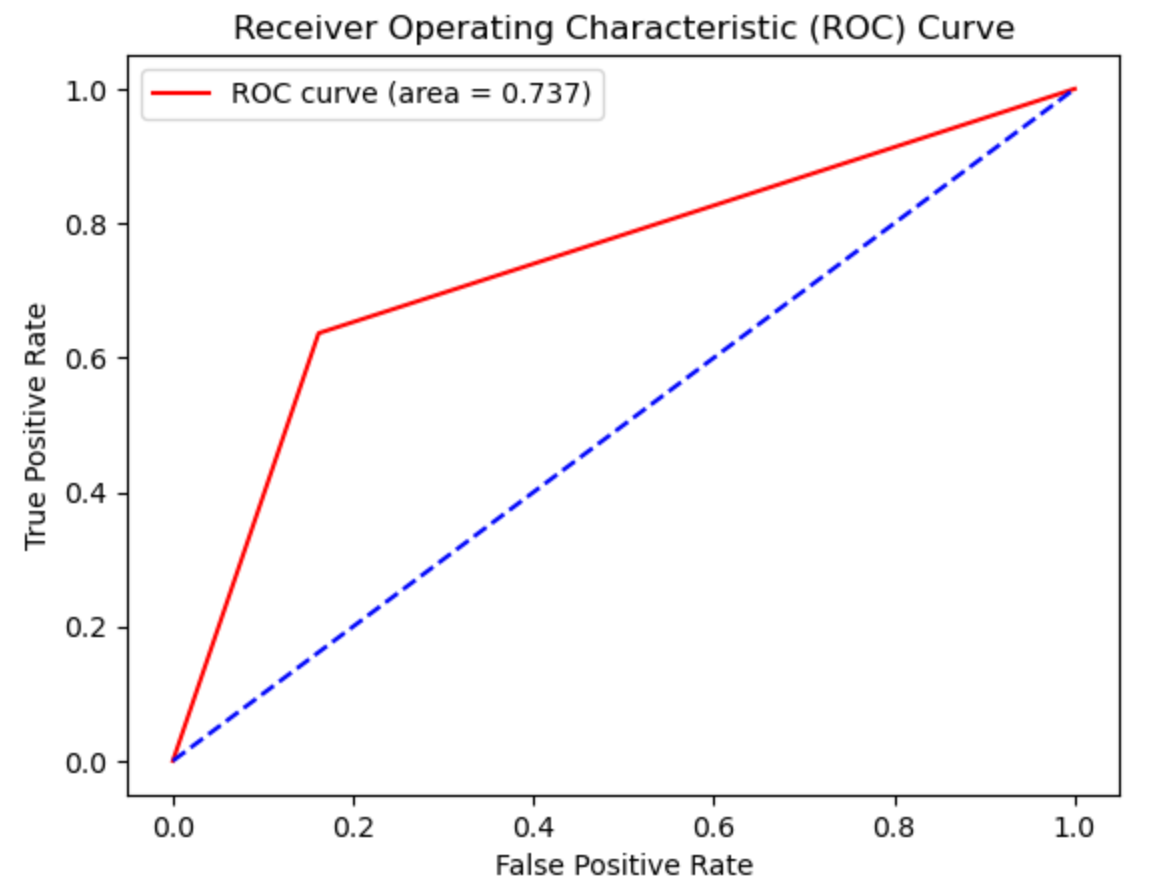
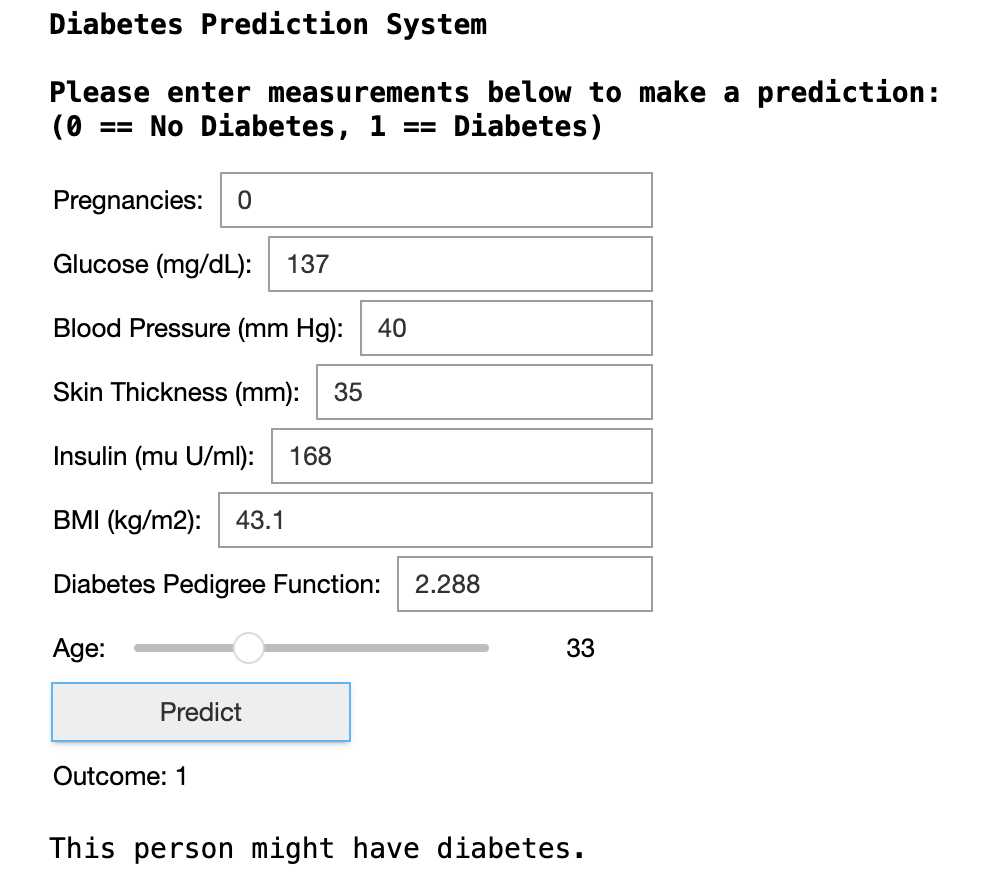


Figure 4. ROC curve with AUC score

Last but not least, to ensure the usability and accuracy of the model’s predictions, a **user interface**, as shown below, is provided for users to try out using the model to make predictions with their new data. After making the predictions, the **outcome** is displayed at the bottom of the interface showing whether or not an individual has diabetes.



**As a result of the validation process**, the model achieved its objectives and success criteria which were set upon the initial of the project, and is ready for deployment into a production environment. However, to maintain its accuracy and performance, the model will be tracked and monitored continuously with user feedback and data re-training to enhance any future updates and refinements.

## Visualizations

Besides the figures illustrated in the previous sections, the descriptive method of the proposal also includes data visualizations such as histogram, scatter plot, pair plot and correlation matrix which are displayed and analyzed below.

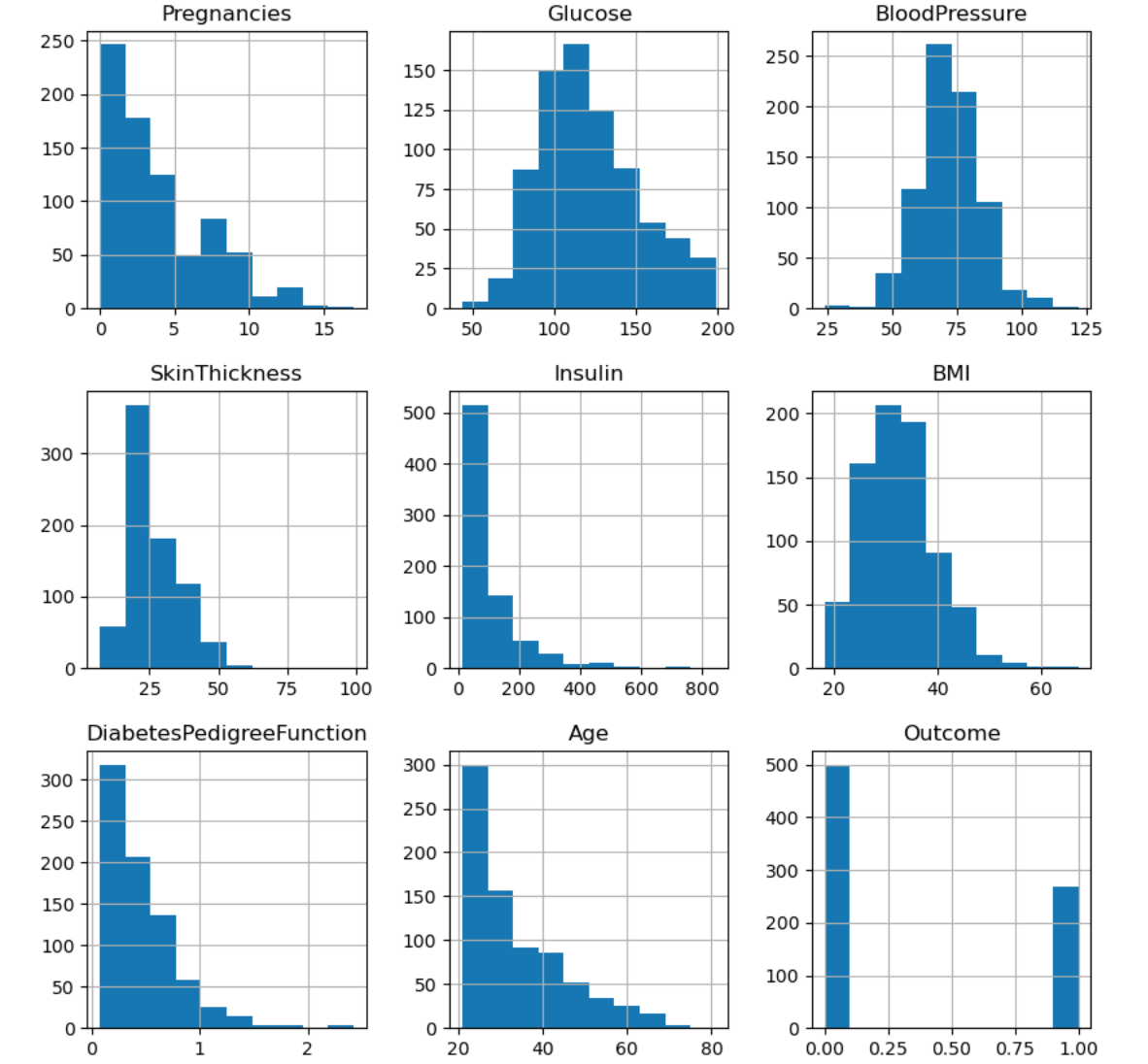


Figure 5. Histogram of each feature

The histogram, as shown in **Figure 5** above, illustrates the distribution of each variable in the dataset. It is often used to check whether the data is normally distributed or skewed to the left or right. The skewness in data provides insight into the patterns or anomalies that might exist in the data. The figure shows that **Pregnancies**, **Skin Thickness**, **Insulin**, **Age** and **Diabetes Pedigree Function** are **skewed to the right**, meaning that the majority of data points are below the average and on the left tail of the graph. This indicates that there might be outliers in the data affecting the model’s performance. Thus, the data is standardized during preprocessing to remove the effect of the outliers.

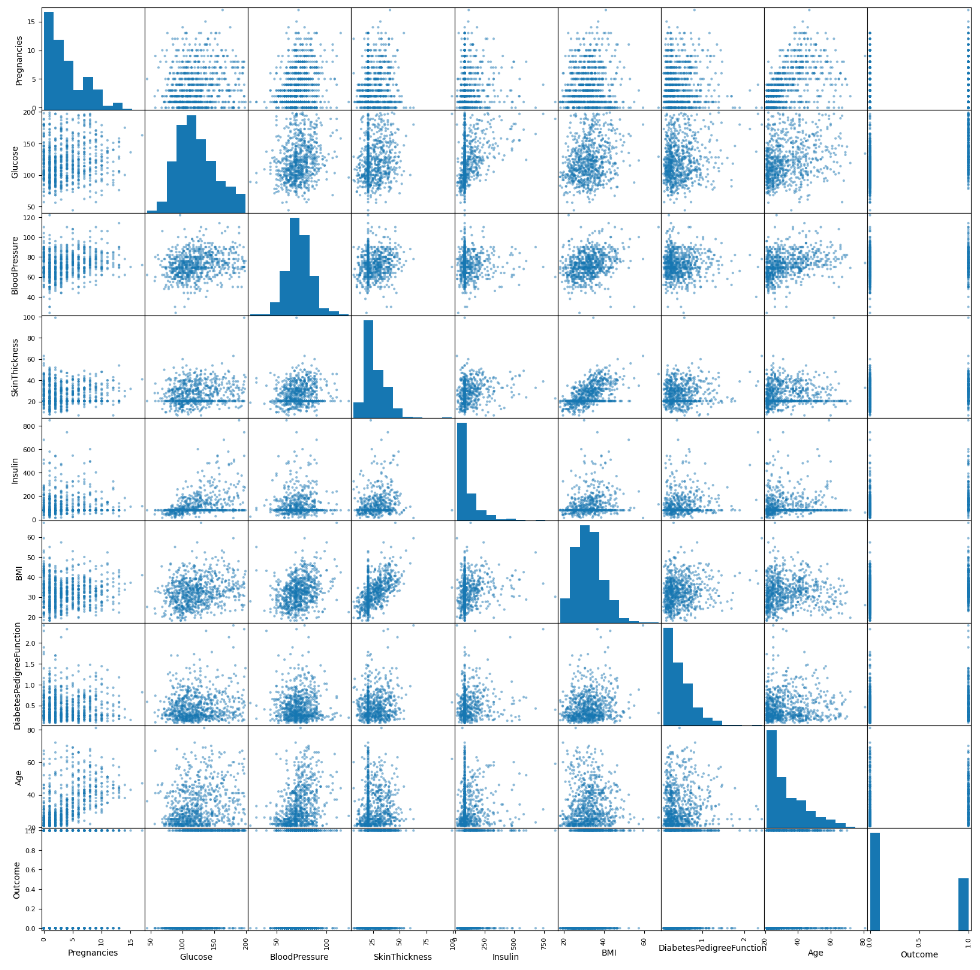


Figure 6. Scatter plot of all features

The scatter plot, as shown in **Figure 6** above, is often used to identify the pairwise relationships between features. The random scattered data points indicate no obvious relationship between features, but if the data points are scattered aligning in a straight line, it means the features are closely related either positive or negative. Referring to the scatter plot above shows that the correlated features consist of **[Pregnancy and Age]**, **[Glucose and Insulin]** and **[Skin Thickness and BMI]** because the data points are closely aligned to a diagonal straight line, indicating a positive correlation in each pair of features.

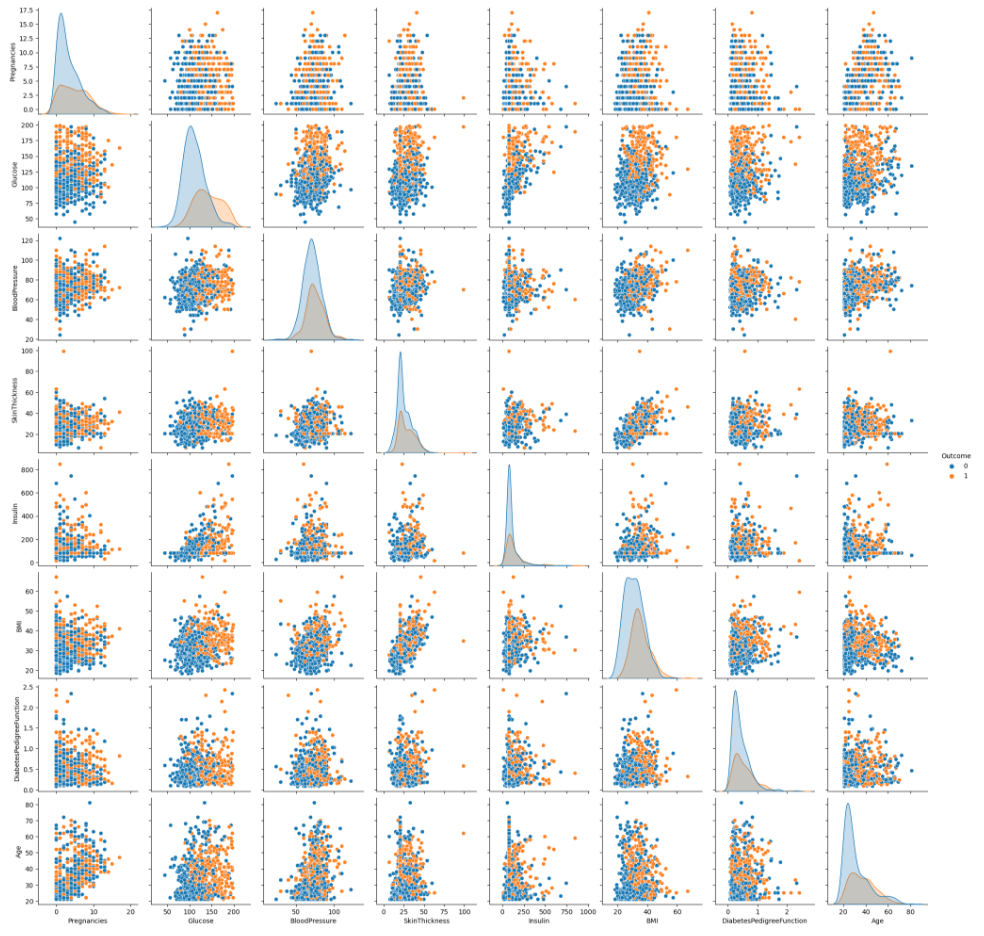


Figure 7. Pair plot of all features

Additionally, the pair plot is used to display the multivariate relationships between all features in the dataset and visualize a categorical variable. It is similar to the scatter plot; however, it visualizes the difference in a variable by showing it in different colors. As shown in **Figure 7** above, the data points are plotted in different colors for each class of the **Outcome** variable, displaying how the data points are scattered differently for each pair of features and how the relationships vary across the Outcome class.

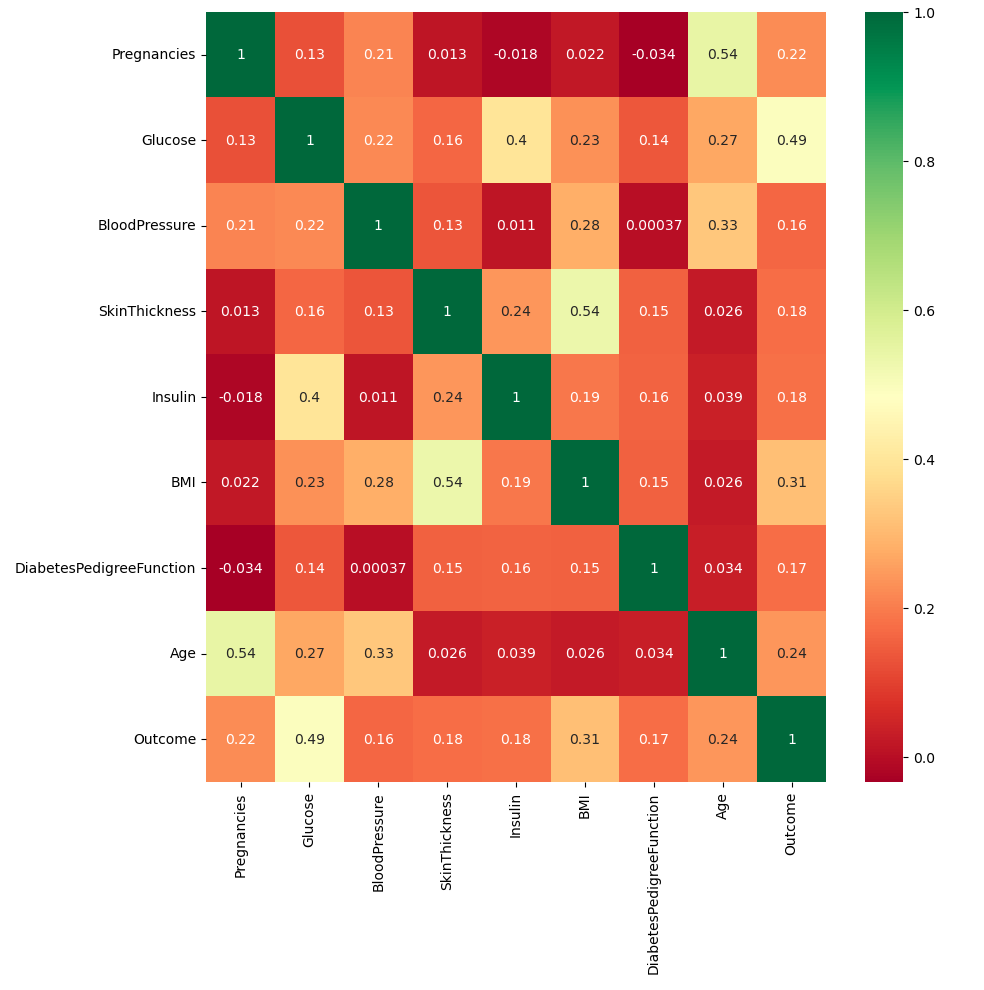


Figure 8. Correlation matrix of all features

Last but not least, the correlation matrix is used to measure the degree of relationship between two variables in the dataset. It tells the dependency level of one variable to another variable based on its correlation value and determines which feature variables are highly correlated to the target variable. This helps in the feature selection process to remove highly correlated feature variables that might affect the model’s performance. The correlation matrix displayed in **Figure 8** above, shows that there is a moderate positive correlation between **Outcome** and several features such as **Glucose (0.49)**, **BMI(0.31)**, **Age(0.24)** and **Pregnancies (0.22)**. However, these features are important measurements associated with diabetes; thus, they will not be removed during the feature selection.

## User Guide

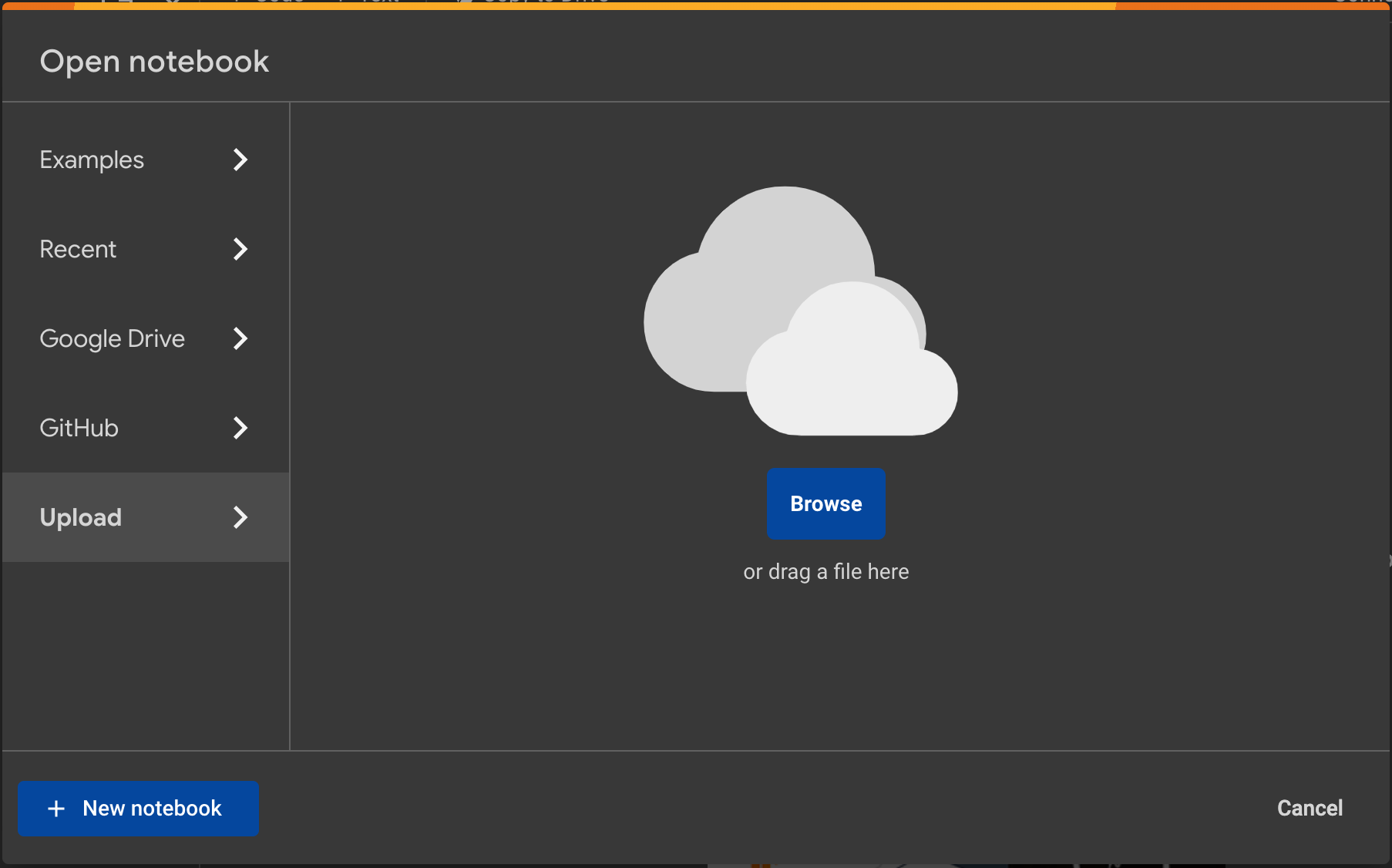
The application needs the Jupyter Notebook file to be executed via Google Colaboratory (Google Colab) on your web browser which is detailed below.

**Installation instructions:**

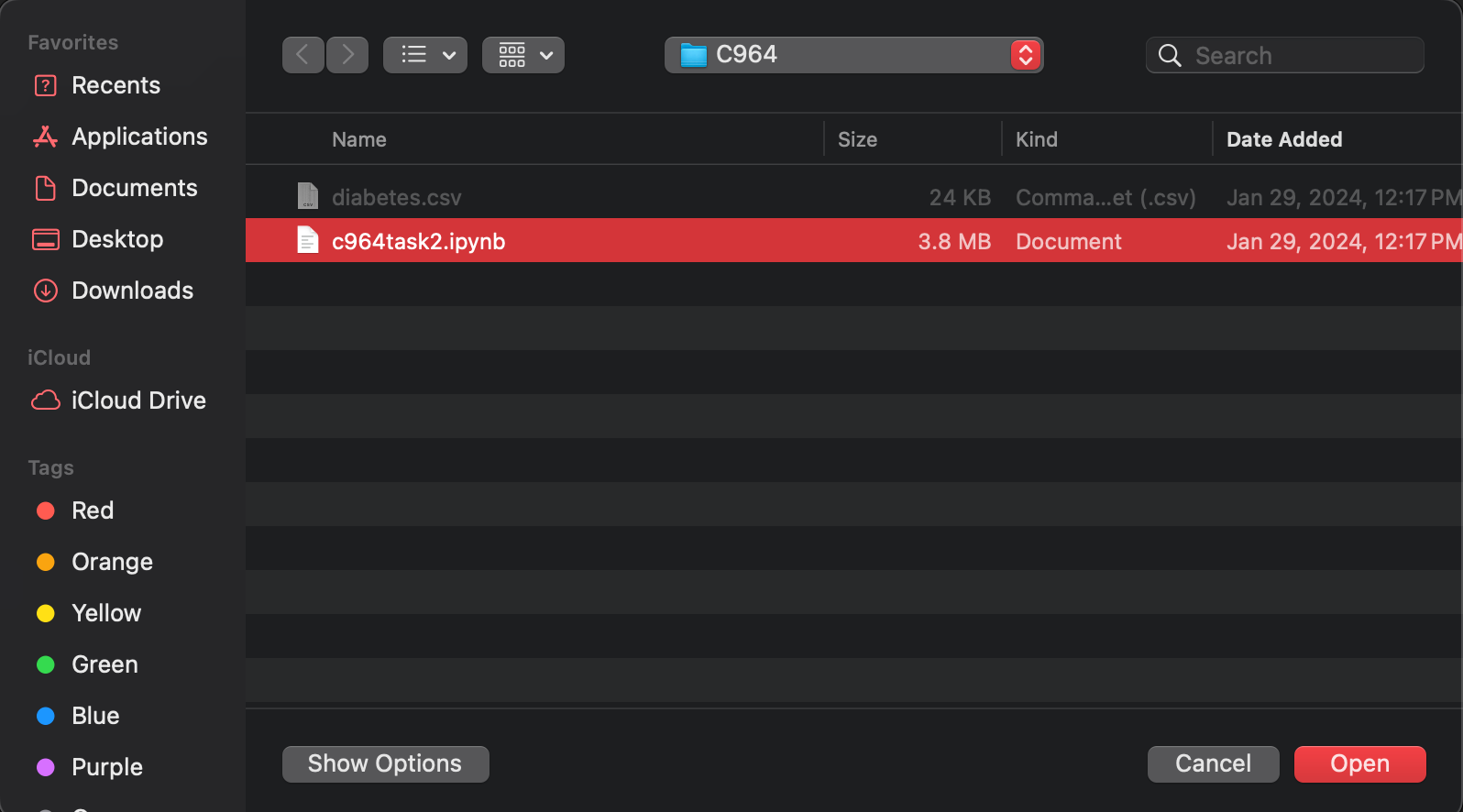
1. First, you need to download the Jupyter Notebook file named **“c964task2.ipynb”** as attached in the proposal.
2. Next, open a new web browser, then sign into your **Google** account.

**\*3A.** To start working with **Google Colab**,

* 1. Go to the URL: <https://colab.research.google.com>
  2. After loading the web page, click on **Upload** on the left column, then click on  **Browse**.

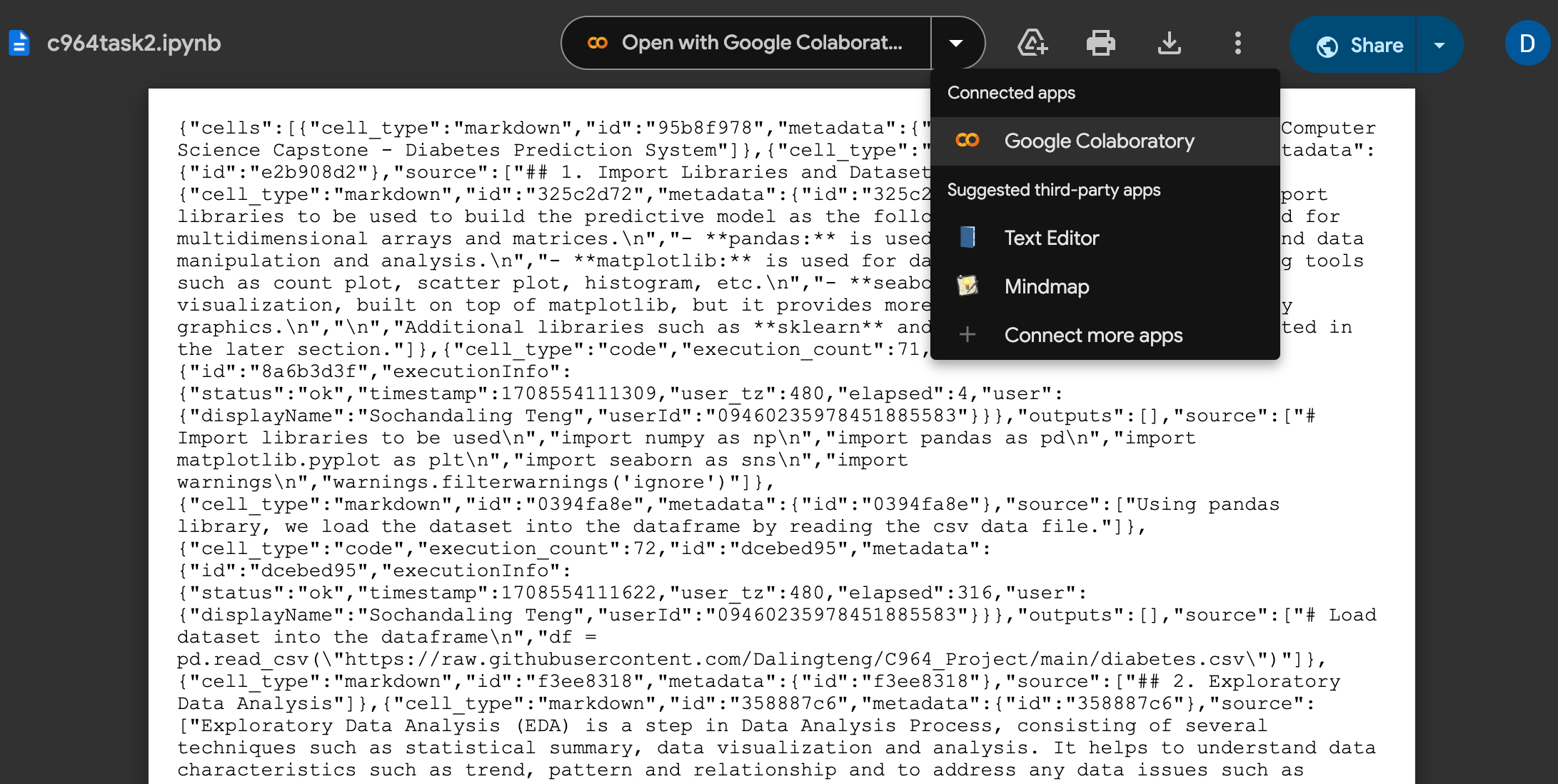


* 1. Next, locate and select the file named **“c964task2.ipynb”** in your file explorer, then click on **Open** to upload the notebook to Google Colab.



**\*3B.** To access **Google Colab** via **Google Drive**,

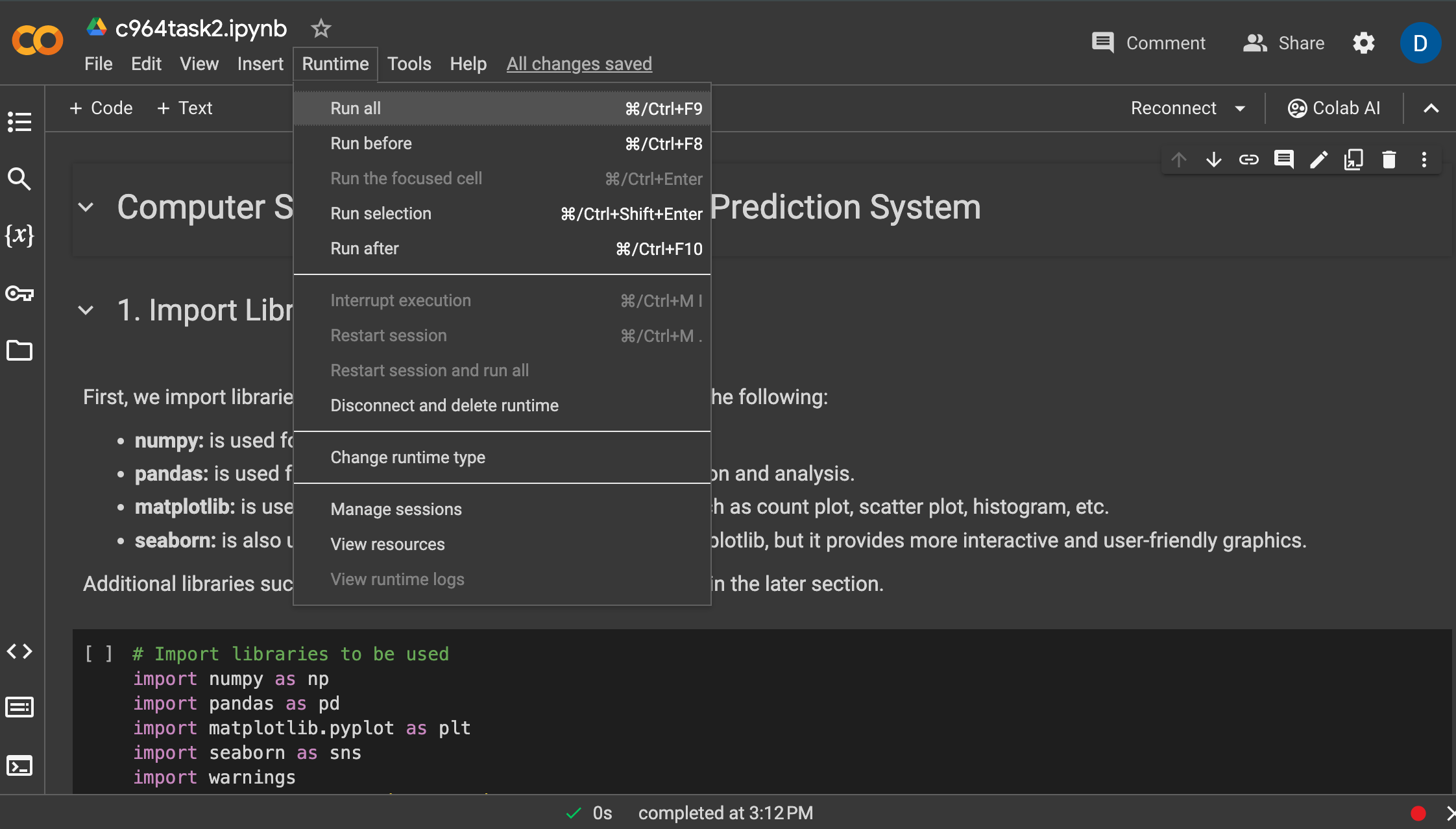
1. Go to the URL below for the notebook hosted in my personal Google Drive: <https://drive.google.com/file/d/1hBiHHUOz5I8xIlitBLZNPqD6T5HWFVPA/view?usp=drive_link>
2. Then, click on **Open with Google Colaboratory** to open the notebook with Google Colab.

****

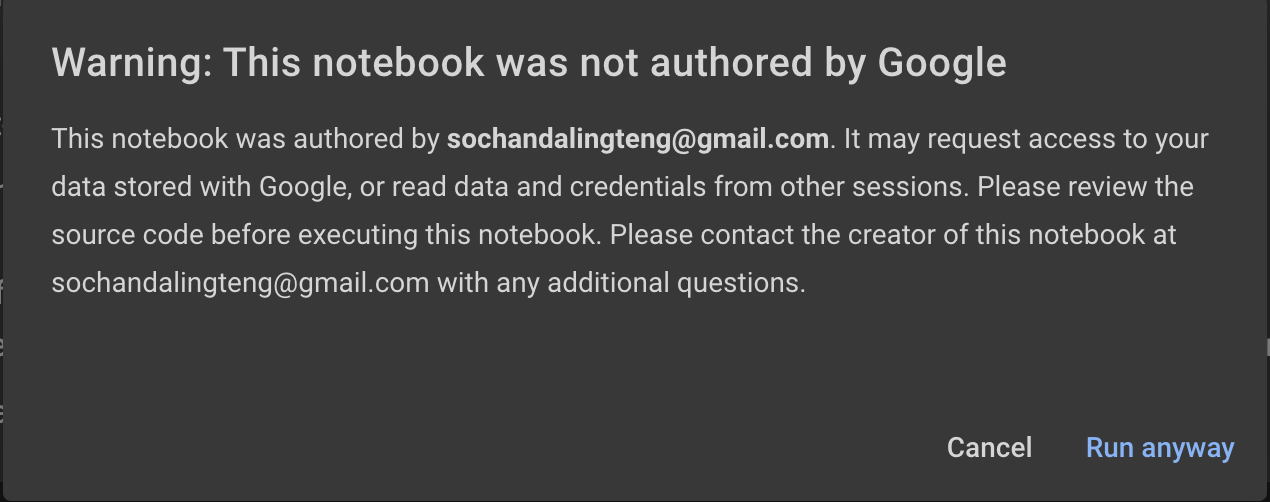
**\*Please note that Step 3A and Step 3B are alternatively done. You can choose one to proceed after completing Step 2.**

**Execution instructions:**

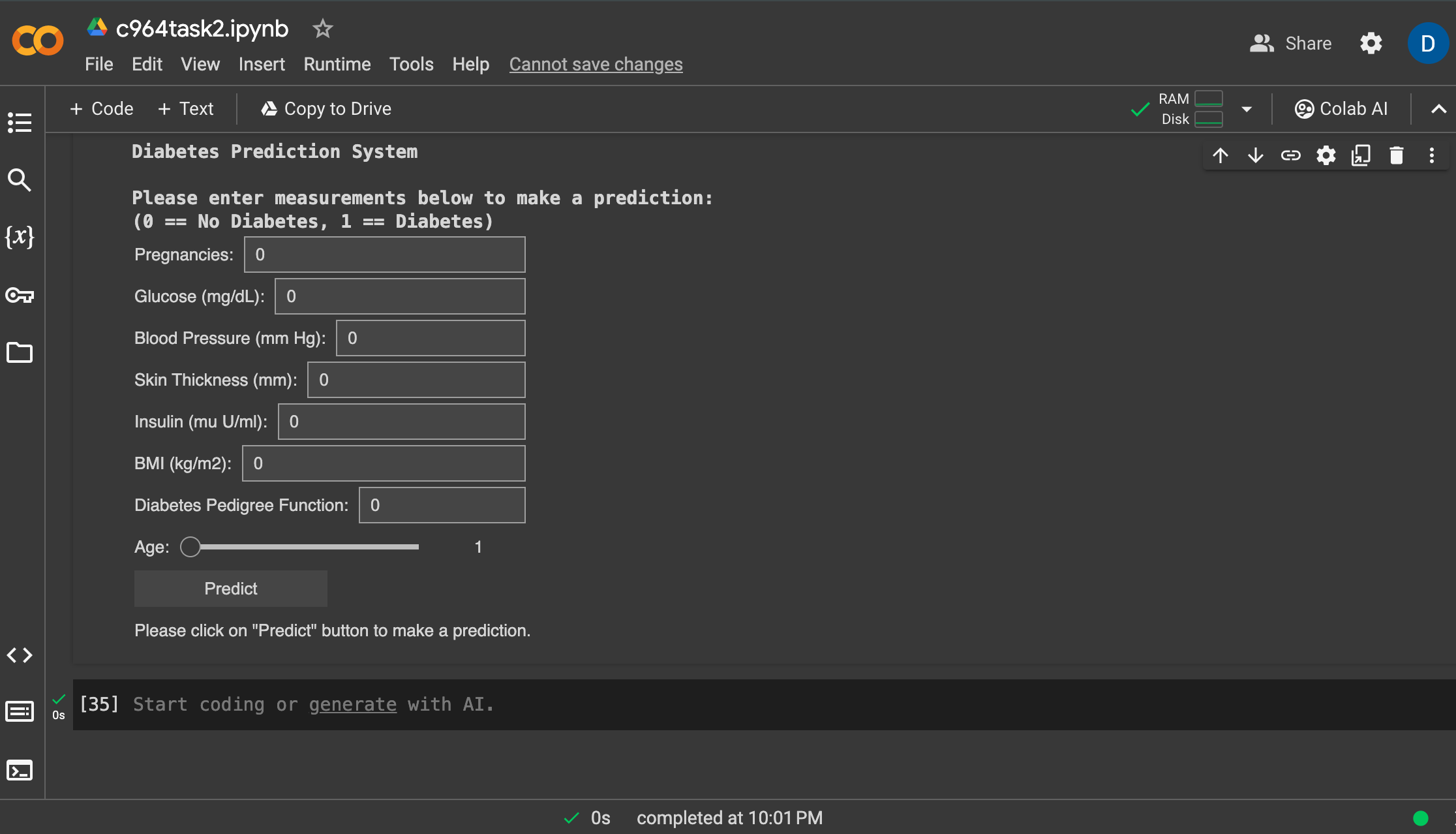
1. In Google Colab, click on **Runtime** in the menu bar, then click on **Run all** to execute all cells.



1. If a **Warning** dialog pops up, click on **Run anyway** to proceed.



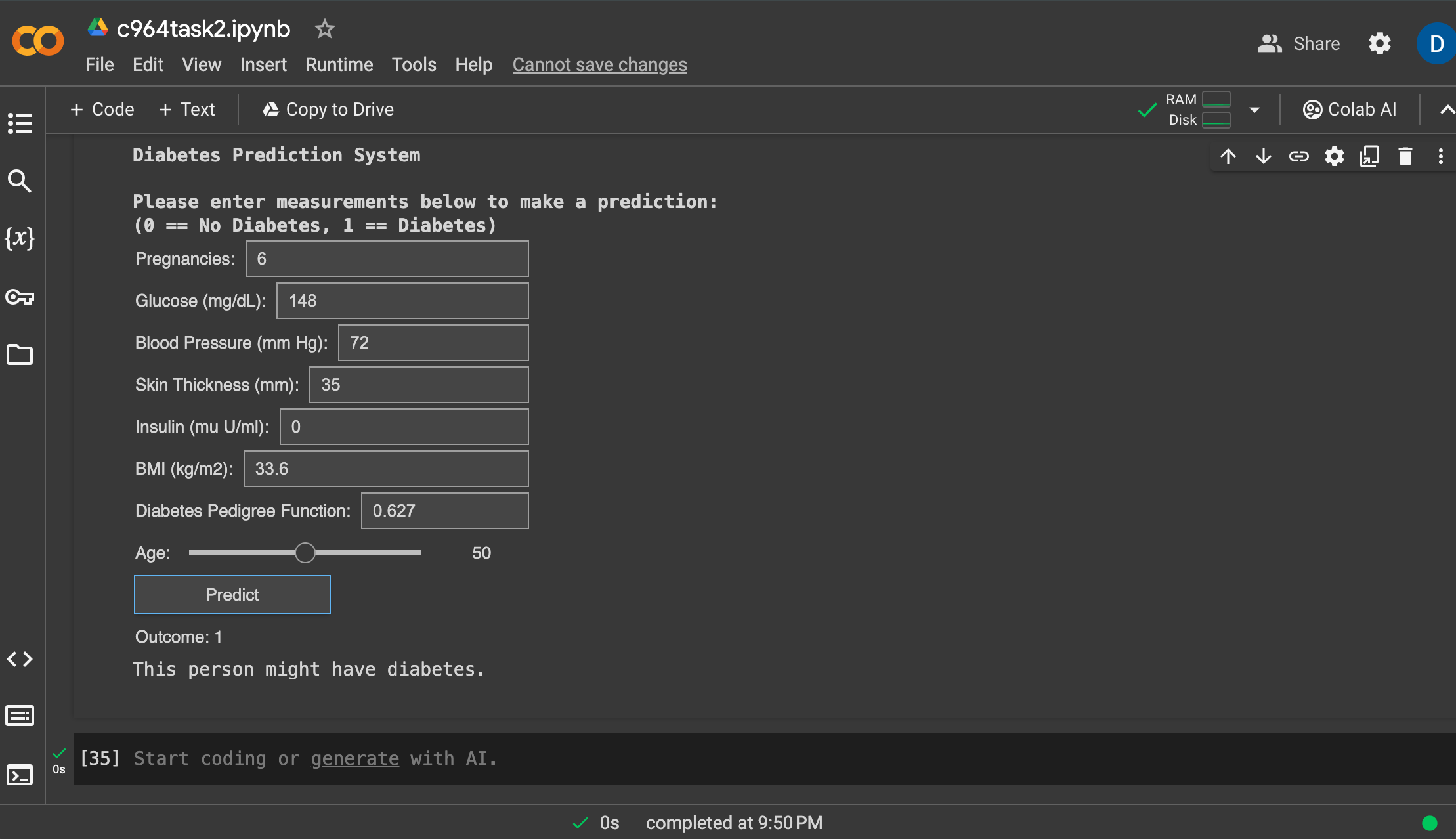
1. After running all the cells, wait a few minutes for the application to finish its execution, then scroll down to the very bottom of the notebook.
2. Navigate to the section **“Diabetes Prediction System”** and follow instructions**.**
3. Next, fill out each field in the form with your data, then click on **Predict** button to generate a result.



Below is a sample of data you can try out to generate results:

****

1. After making a prediction, the result generates an **outcome** and prints out a **message** about whether the person might or might not have diabetes.



# References

Centers for Disease Control and Prevention. (n.d.). *National Diabetes Statistics Report.* U.S. Department of Health and Human Services. <https://www.cdc.gov/diabetes/data/statistics-report/index.html>

Chumbar, S. (September 21, 2023). *The CRISP-DM Process: A Comprehensive Guide.* Retrieved from <https://medium.com/@shawn.chumbar/the-crisp-dm-process-a-comprehensive-guide-4d893aecb151>

Leslie O. Schulz, Peter H. Bennett, Eric Ravussin, Judith R. Kidd, Kenneth K. Kidd, Julian Esparza, Mauro E. Valencia; *Effects of Traditional and Western Environments on Prevalence of Type 2 Diabetes in Pima Indians in Mexico and the U.S*.. Diabetes Care 1 August 2006; 29 (8): 1866–1871. <https://doi.org/10.2337/dc06-0138>

Shipe, M. E., Deppen, S. A., Farjah, F., & Grogan, E. L. (2019). Developing prediction models for clinical use using logistic regression: an overview. *Journal of thoracic disease, 11*(Suppl 4), S574–S584. <https://doi.org/10.21037/jtd.2019.01.25>

UCI Machine Learning. (October 6, 2016). *Pima Indians Diabetes Database*. Kaggle. <https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database/data>