School of Computing and Engineering

Final Year Project

Project Report

Name: Priscilla Philby Oommen

Project Title: Prediction of Type 2 Diabetes using Machine Learning

Date: 19/05/2024

Abstract

T2D is a metabolic disorder that results from the body’s inability to make enough insulin or to properly utilize the insulin produced in the body. It is the most common type of diabetes. It is a chronic disorder that often develops gradually, with symptoms happening after the disease has advanced enough. Undiagnosed type 2 diabetes may cause nerve and kidney damage, heart and blood vessel disease, slow healing of wounds, hearing impairment and several skin diseases. So, Early detection of diabetes is essential to have a healthy life. Machine learning algorithms can detect individuals at risk of acquiring diabetes based on their health data, enabling prompt intervention and preventative measures to slow down the development of the disease. “Prediction of Type 2 Diabetes” is a machine learning project in which a predictive model analyses user-entered input and compares it with data present in the dataset and predicts whether the person is diabetic or not via an interactive GUI. Data preprocessing, data analysis, data cleaning, data visualization, and feature selection are performed, predictive models are built using ML algorithms such as Logistic Regression, Decision Tree, Random Forest, and XGBoost. The performance of the models is evaluated using performance matrix. XGB model has the highest accuracy rate and was chosen for further development. Gradio Interface is used for GUI development. Users can enter medical information to determine if they have type 2 diabetes or not, and the ML model almost accurately predicts and classifies if the users have type 2 diabetes or not. It also predicts if a user has a risk of having prediabetes. This whole project has been implemented using Google Colaboratory.

TABLE OF CONTENTS

1 Figures and Abbreviations

1.1 List of Figures

1.2 List of Abbreviations

i. T2D – Type 2 Diabetes

ii. ML – Machine Learning

iii. LR – Logistic Regression

iv. DT – Decision Tree

v. RF – Random Forest

vi. XGB – Extreme Gradient Boosting

vii. BMI – Body Mass Index

viii. GUI – Graphical User Interface

ix. Google Colab – Google Colaboratory

2 Introduction

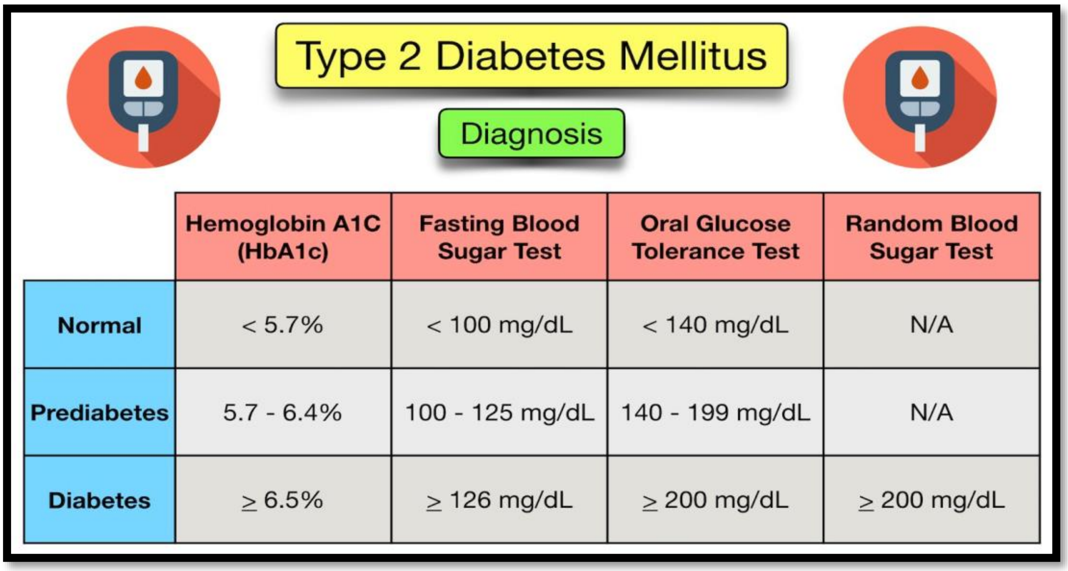
Diabetes Mellitus is a set of metabolic disorders defined by persistently elevated blood sugar levels. This occurs when the body does not create enough insulin (Type 1 diabetes), cells do not respond to the insulin produced (Type 2 diabetes), or a combination of the two. Insulin is a hormone made by the pancreas that helps regulate blood sugar levels by enabling the absorption of glucose into cells. Insulin stimulates intake, storage, and utilization of glucose. [1] It is the leading cause of kidney failure, heart attacks, strokes, and lower-limb amputations. Diabetes mellitus type 1 and type 2 are chronic conditions. Prediabetes and gestational diabetes are the two reversible types of diabetes. [8] Type 1 diabetes is an autoimmune disease where the immune system targets and destroys insulin-producing cells in the pancreas for unclear reasons. Approximately ten percent of diabetics have Type 1. It is typically diagnosed in children and young adults, but it may develop at any age. Prediabetes is the condition that occurs before Type 2 diabetes. In prediabetes, the blood glucose levels rise more than normal, however not sufficiently high to be clinically diagnosed as Type 2 diabetes. Gestational diabetes is a kind of diabetes that develops during pregnancy. Gestational diabetes normally subsides after pregnancy. However, having gestational diabetes increases your risk of acquiring Type II diabetes later on in life.

2.1 Type 2 Diabetes

[1][8]Type 2 diabetes is a metabolic disorder that results from the body's inability to make enough insulin or to properly utilize the insulin produced in the body. Over 95

percent of diabetics have type 2 diabetes. Type 2 diabetes problems can be avoided with early identification. It mostly affects adults, but it is becoming more common in children and adolescents as obesity rates rise. In type 2 diabetes, the symptoms can be mild and may take many years to be noticed. Signs and symptoms include excessive thirst, unintentional weight loss, frequent urination, slow healing of wounds, increased hunger, fatigue, blurred vision, and numbness. The risk factors for developing T2D are of two types, modifiable risk factors and non-modifiable risk factors. Modifiable risk factors are obesity, overweight, physical inactivity, Unbalanced diet, and Prediabetes. Non-modifiable risk factors are having family history of diabetes, age more than 45 years, people who have ethnicity of South Asian, African, and Hispanic, low birth weight, previous pregnancy with gestational diabetes and having a history of PCOS or other metabolic diseases.

[1][8]A healthcare professional may diagnosis T2D via a variety of different laboratory tests, like a random blood sugar test (which can be taken randomly anytime, even though you ate food or not, a HbA1C test (which measures your average blood sugar levels over a period of three months), a fasting blood sugar test (which evaluates your blood sugar levels when you), and a glucose tolerance test (in which you drink a glucose-containing liquid and your blood glucose levels are recorded over time).



Type 2 Diabetes Mellitus

Diagnosis

	Hemoglobin A1C (HbA1c)	Fasting Blood Sugar Test	Oral Glucose Tolerance Test	Random Blood Sugar Test
Normal	< 5.7%	< 100 mg/dL	< 140 mg/dL	N/A
Prediabetes	5.7 - 6.4%	100 - 125 mg/dL	140 - 199 mg/dL	N/A
Diabetes	≥ 6.5%	≥ 126 mg/dL	≥ 200 mg/dL	≥ 200 mg/dL

Figure 2-1:Diagnosis of T2D [15]

[1][8]Type 2 Diabetes is managed and treated by keeping blood glucose levels within a target range using a combination of lifestyle changes, oral or injectable medicines, and, if needed, insulin therapy. Regular monitoring, a nutritious diet, physical activity,

and competent medical treatment are all essential for successfully managing diabetes and avoiding complications.

2.2 Machine Learning Classification Algorithms

[2] Machine learning (ML) uses statistical models and algorithms that allow computer systems to uncover patterns in huge volumes of data, then utilizes a model to recognize those patterns for making predictions or descriptions about new data.
Two types of techniques are used in machine learning: unsupervised learning, which looks for intrinsic structures or hidden patterns in input data, and supervised learning, that trains a model with known data from the input and output to predict future outputs. Unsupervised learning involves Clustering. Supervised learning involves Classification and Regression algorithms.

[2] Machine learning makes use of complex algorithms that find patterns in data in order to create models. These models are useful for classifying data and for forecasting and prediction purposes. The principal attributes of using machine learning are automated identification of patterns, estimating the most likely results, generation of useful information, and the possibility of analyzing substantial amounts of data.
This project deals with a classification problem, that is, classifying a person is diabetic or not. So, it is a machine learning project utilizing classification algorithms to predict whether a person has Type 2 Diabetes or not, based on certain important features like Blood glucose level, HbA1c level, BMI, etc. ML algorithms selected for this project are Extreme Gradient Boosting (XGB), Random Forest, Decision Tree, and Logistic Regression. They are state-of-the-art prediction algorithms with high accuracy rate.

The most common algorithms used for building predictive models are:

1. **Decision Tree**
   [16]Decision tree divides the data into subsets according to the most important characteristics in a recursive manner, modeling decision-making processes. Each internal node of the decision tree algorithm indicates a decision made based on a particular feature, and each leaf node represents the expected value or outcome. The result is a structure like a tree. The top node of the tree is the initial query or feature that best divides the data into two sets. The nodes in the middle reflect further queries or features, resulting in branches based on various situations. The final nodes at the bottom (leaves) offer the expected outcome or class. The algorithm determines the optimal feature and condition to partition the data at each node. Maximizing homogeneity within each subgroup is the aim. For classification problems, this is commonly measured using Gini impurity or information gain. Applications of decision trees in healthcare is to assist in diagnosing diseases. Decision trees are useful because they can handle both numerical and category variables and are easy to interpret.

2. **Random Forest**

[16]An ensemble (compilation) of decision trees makes up a Random Forest. Randomness is added in terms of the data samples and the attributes used for splitting at each node. Each tree is trained independently using a random subset of the training data. Unlike individual decision trees, Random Forests work as an ensemble, which helps minimize overfitting. The model's ability to generalize successfully to new data is facilitated by the diversity among the trees. Due to the independent training of every tree in the ensemble, Random Forests are easily parallelized. They are therefore scalable and effective for big datasets. Robust and popular machine learning algorithms, Random Forests are renowned for their robustness, versatility, and ability to handle intricate data interactions. So, it is often chosen for data analysis and machine learning projects because of its high performance and ease of use.

3. **XGBoost**

[2]Gradient Boosting is an ensemble learning technique that combines the predictions of several weak models, usually decision trees, to create a powerful predictive model. Belonging to the gradient boosting family, XGBoost, also known as Extreme Gradient Boosting, is a potent and effective machine learning technique. It is intended for both regression and classification problems. During training, XGBoost comes with built-in functionality for managing missing values in the dataset. "Pruning" is a strategy that the algorithm employs when creating new trees. Pruning aids in tree size control, which lowers overfitting and enhances generalization. It is hence effective for computationally demanding tasks and vast datasets. XGBoost algorithm's outstanding predictive performance, versatility, and robustness have made it a popular choice in both research and IT industry.

4. **Logistic Regression**

[16]Logistic regression is a classification ML algorithm that is used to predict the likelihood that an instance will fall into one of two classes in binary categorization. It models the likelihood of an event happening using a logistic (sigmoid) function. The predicted probability is obtained by transforming the linear combination of input features and their corresponding weights using the sigmoid function. Logistic regression is a popular algorithm because of its ease of use, interpretability, and efficiency and it is an essential component of several machine learning pipelines, particularly those in which the objective is to predict binary outcomes.

2.3 Project Overview

“Prediction of Type 2 Diabetes” is a machine learning project in which a predictive model analyses user-entered input and compares it with data present in the dataset and predicts whether the person is diabetic or not via an interactive GUI. Dataset 1 and Dataset 2 are Kaggle datasets, namely PIMA Indian Diabetes Dataset and Diabetes Prediction Dataset are used in this project. Data analysis and data visualization are performed, predictive models are built using ML algorithms such as Logistic Regression (LR), DT, RF, and XGB, and development are implemented using Google Colaboratory because it is a free online cloud based Jupyter Notebook service that requires no setup to use and provides free access to powerful computing resources, like CPU, GPU, TPU. It is efficient, ease of use, accessible in any IT device with an internet connection from anywhere worldwide.

Data Analysis: Data Preprocessing, Exploratory data analysis (EDA), Data Cleaning, Feature selection is done. Necessary libraries such as Pandas and NumPy are imported to Google Colab. The dataset is loaded into a Pandas Data Frame. Exploratory data analysis (EDA) is performed to understand the structure, summary statistics, and distributions of the data. Missing values, null values, duplicate values in the dataset are checked and handled. In Dataset 2, Data preprocessing steps such as encoding categorical variables using LabelEncoder() are performed only for categorical data. Feature scaling was not performed because the dataset contains medical data. Outliers can be present in the data. It is not removed from the data because it is medically possible values and are not extreme or impossible values. In Dataset 1, QuantileTransformer() is used for feature scaling and removing unnecessary outliers from the dataset. It doesn't change much in the data even if it is applied but it will help improve the accuracy of ML models. Class imbalance was found for target feature in both datasets. The majority class(0) that, absence of diabetes, has outnumbered the minority class(1), that, presence of diabetes. It was handled using SMOTE technique before splitting the datasets into x and y.

Data Visualization: Python libraries like Matplotlib and Seaborn were used for data visualization. Histogram, box plots, count plot, pie chart, donut chart and heatmap were plotted to visualize relationships between various features, display important insights or interesting findings and identify patterns. Outliers in dataset is identified by using boxplot to visualize the target feature of the datasets. Class imbalance is identified in target variable by using count plot.

Building Predictive Models: Both datasets are divided into x and y and then split into 70% training and 30% testing set using Train Test Split. Python libraries such as Scikit-learn are imported for building machine learning models. These predictive models are then created and trained on the training data using ML algorithms like Linear Regression, Decision Trees, Random Forests, and XGBoost. Evaluation metrics like accuracy, precision, recall, f1-score, confusion matrix are used to evaluate the performance of the models. This is done because my project deals with a classification problem. The predictive models are compared with respect to the performance matrix. The model with the highest accuracy was XGBoost algorithm in both datasets. XGBoost algorithm achieved an accuracy rate of 76% in Dataset 1 and an accuracy rate of 97% in Dataset 2.

GUI Development: XGBoost algorithm and Dataset 2 is chosen for further development for creating a graphical user interface (GUI) using Gradio Interface in Google Colab for T2D prediction because XGBoost has highest accuracy and Dataset 2 has a special feature called ‘HbA1c level’ which is not present in Dataset 1. Actually, HbA1c test is a diagnostic test done to classify a person as healthy, prediabetic or diabetic by a medical professional.

2.4 Relevance and Rationale of Study

Type 2 diabetes is a chronic disorder that often develops gradually, with symptoms happening after the disease has advanced enough. Undiagnosed type 2 diabetes may cause nerve and kidney damage, heart and blood vessel disease, slow healing of wounds, hearing impairment and several skin diseases. Early detection of diabetes is essential to have a healthy life. Machine learning algorithms can detect individuals at risk of acquiring diabetes based on their health data, enabling prompt intervention and preventative measures to slow down the development of the disease.

2.5 Structure of dissertation

The chapters of the dissertation are organized as follows:

Chapter 1 delivers the required list of the figures and tables that are utilized in the report.

Chapter 2 gives a basic overview and introduction to the use of machine learning in the prediction of Type 2 Diabetes. The process for predicting Type 2 Diabetes is also covered in this section of the chapter, along with the machine learning steps of feature scaling, data preprocessing, data cleaning, data visualization, and creating predictive models using ML algorithms.

Chapter 3 reviews recent research publications that use machine learning to predict type 2 diabetes. The literature review study demonstrates how various ML algorithms are used to build a wide variety of models. This section of the chapter goes on to discuss the findings and accuracy of the reviewed research papers as well as their limitations. The research gaps and questions that have been identified are covered in this section.

Chapter 4 describes the aim and objectives of the project are outlined in the section that follows in the chapter, which also includes the reasons behind selecting it.

Chapter 5 explains the methodology that was chosen for this project. This section discusses the CRISP-DM technique and explains why this project was a good fit for it. This chapter also attempts to address the research approach, which is a quantitative analysis.

Chapter 6 explains the project's technical implementation and design, including the machine learning process, the tools and techniques employed, and the GUI development. The various steps taken to build the project are detailed in this section, including data collection, data preprocessing, data cleaning, feature selection, model building, model training, evaluation matrix, GUI development.

Chapter 7 explains the analysis and results in depth. Findings such as the interpretation of data visualizations and the comparison of various machine learning models utilizing evaluation matrices that include f1 score, AUC value, accuracy, precision, and recall will be included. It also displays a graphical user interface (GUI) where users can enter medical information in order to find out if they have Type 2 diabetes or not.

Chapter 8 explores the project's discussions and conclusion. This chapter additionally addresses the project's limits and provides recommendations for future works.

Chapter 9 lists the references used in this project.

Chapter 10 depicts the appendix, which contains Weekly Progress Forms 1 and 2.

3 Literature Review

When I was researching recent research papers about the topic, Prediction of Type 2 diabetes using machine learning techniques, I came across many research papers. In most of them, the dataset used for the project was PIMA Indian Diabetes Dataset. I have chosen five research papers to do the literature review.

(Tigga and Garg, 2020) In India, more than 30 million people have diabetes, and many more are at risk. Therefore, diabetes and health problems it causes must be prevented with early detection and treatment. In this research paper, Two datasets are used. Questionnaire dataset and PIMA Indians dataset. The purpose of this study is to evaluate a person's risk of developing diabetes based on their lifestyle and family history. Several machine learning algorithms such as Random Forest, K-Nearest Neighbour, Support Vector Machine, Decision Tree, Naive Bayes, Logistic Regression were used for predicting the risk of Type 2 diabetes since they are exceedingly accurate, a critical component for anyone working in the health field. People can self-evaluate their risk of diabetes once the model has been trained with a high degree of accuracy. 952 cases have been gathered for the project via an offline and online questionnaire that included 18 inquiries about family history, lifestyle, and health. The same algorithms were applied to Pima Indian Diabetes dataset. The results show that the accuracy of Random Forest is the highest for questionnaire dataset, i.e., 94% and for PIMA dataset is 75%. This result can be used in future to predict any other ailment. This study still holds a scope for further research and improvement, including other machine learning algorithms to predict diabetes or any other disease.

(Nadaf et al., 2023) In this work, supervised machine learning algorithms are used to detect diabetes based on clinical data. Models like Decision Tree, Naive Bayes, k-Nearest Neighbour, Random Forest, Gradient Boosting, Logistic Regression, and Support Vector Machine are trained on a variety of datasets. Effective preprocessing methods like label encoding and normalization increase the model's accuracy. Risk indicators are ranked according to a range of feature selection techniques. To evaluate the model's performance, two distinct datasets are subjected to thorough testing. Depending on the dataset and machine learning method, accuracy increases can be anywhere between 2% and 12% higher than in earlier research. The results show that SVM is better than other ML algorithms in assessing how well it predicted an individual's diabetes. Accuracy rate of SVM is 79.5%. The model has been integrated into a web-based application using Flask in Python, which will be made available via Docker. According to the study, diabetes may be reliably and efficiently predicted by combining machine learning-based categorization with an adequate data preparation pipeline, which would facilitate an early diagnosis and better health outcomes. In the future, the ML model will be able to look at a bigger, more comprehensive dataset of patients with other characteristics so that actual users may assess the accuracy.

(AKMEŞE, 2022) Using diagnostic measurements from PIMA Indians Diabetes Dataset from the National Institute of Diabetes and Digestive and Kidney Diseases, this study aims to predict if a patient has diabetes. As the input variable, eight distinct patient factors were selected, and it was determined whether the patient had diabetes or not. Out of the 768 records that were reviewed, 268 (34.9%) had diabetes and 500 (65.1%) were in good health. A total of ten distinct machine learning algorithms have been used to predict the possibility of diabetes. Random Forest, Gradient Boosting, XGB, LGBM, Decision Tree, AdaBoost, Support Vector Machine, Logistic Regression, KNN and Naive Bayes algorithms have been applied to predict diabetic status. With an accuracy of 90.1%, the Random Forest algorithm proved to be the most effective approach. Other algorithms' accuracy percentages range from 89% to 81%. This research provides a machine learning prediction tool for diabetic patients that is highly precise. The study's proposed model can be useful for detecting diabetes early on. New patient data can be added to train the model in future studies. A better result can be obtained for estimation as the number of data increases.

(Dey, Hossain and Rahman, 2018) In this study, the researchers have created a framework that can determine whether a patient has diabetes or not. The primary goal of this investigation is to develop a web application using a powerful machine learning algorithm that has higher prediction accuracy. They made use of the Pima Indian benchmark dataset, which can forecast the onset of diabetes based on diagnostic methods. Min Max Scaler (MMS) is used for Normalization. ML models like SVM, KNN, Naive Bayes and ANN. ML libraries such as sklearn, NumPy, matplotlib, pandas, and tensorflow.js are used. The development of an interactive web application for diabetes prediction is driven by the remarkable gain in accuracy that the Artificial Neural Network (ANN) shows, with a prediction rate of 82.35%. In future, a deep learning model can be used, and a Location based Dataset can be prepared from real medical data for the successful prediction of diabetes disease.

(Ahmed et al., 2021) Finding efficient machine-learning-based classifier models to use clinical data to identify diabetes in people is the aim of this study. Decision tree (DT), Naive Bayes (NB), k-nearest neighbour (KNN), Random Forest (RF), Gradient Boosting (GB), Logistic Regression (LR), and Support Vector Machine (SVM) are among the machine learning methods to be trained using many datasets in this article. The researchers applied efficient pre-processing methods, such as normalization and label-encoding, to increase the models' accuracy. Furthermore, they determined and ranked a variety of risk indicators using different feature selection techniques. Numerous tests have been run on two distinct datasets to evaluate the model's performance. When their model is compared to some current research, the findings indicate that, depending on the dataset and the chosen ML method, the suggested model provides a higher accuracy of 2.71% to 13.13%. Ultimately, SVM outperforms the other algorithms and provides better accuracy of 81.13%. Flask web development framework in Python was used to integrate this model into a web application. The findings of the study imply that diabetes can be efficiently and accurately predicted with the use of ML-based classification and an adequate preprocessing pipeline on clinical data. Additional research and development can be done using a variety of deep learning methods. In the future, we will examine a larger dataset for patients with additional attributes for improved accuracy.

3.1: Identified Research Gap

1. The dataset used for data analysis and machine learning in the research papers is the PIMA Indian Diabetes dataset, which is a benchmark dataset. For my project, I will used this dataset along with another Kaggle dataset called Diabetes Prediction Dataset:
→ It is a relatively new dataset with medical and demographic data (features: Age, Gender, Body Mass Index (BMI), Hypertension, Heart disease, Smoking history, HbA1c level, Blood glucose level, Diabetes).
→ It is unique from Pima Indian Diabetes Dataset (Features: No. of Pregnancies, Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, Age, Outcome) because it has features like HbA1c level which plays a crucial part in detecting diabetes. It also has a feature 'smoking_history' which is absent in Pima Indian Diabetes Dataset. So, it will be used for GUI development.
→ In the real world, HbA1c blood test results help doctors to diagnose a person as healthy, prediabetic or diabetic. A1C level below 5.7% is considered normal. If it is between 5.7% and 6.4%, it is classified as prediabetes. If A1C level is above 6.5% it is classified as diabetes. The possibility that a person may acquire type 2 diabetes increases with his/her A1C level.
→ Pima Indian Diabetes Dataset concentrates on a particular gender and population group, that is, women of Pima Indian heritage whereas Diabetes Prediction Dataset contains information applicable to the general public.

2. Implementing a user-friendly GUI that will classify a person as diabetic or non-diabetic according to certain diagnostic measurements entered by the user.

3. A comparison study using evaluation matrix among many ML algorithms such as Decision Tree, Random Forest, XGBoost and Logistic Regression on two datasets such as Pima Indian Diabetes Prediction Dataset and Diabetes Prediction Dataset . All the model's accuracy, precision recall, f1-score values can be compared against each other.

3.2: Research Questions

Q1. How can ML model be built to predict whether a person has type 2 diabetes or not?

Q2. Can ML classification model accurately predict diabetes based on certain diagnostic measurements included in the dataset?
Q3. How is the ML model for diabetes prediction built in Google Colaboratory incorporated as a GUI?

4 Aim and Objectives

4.1: Aim

To accurately predict whether a patient has type 2 diabetes or not, using Machine Learning Classification Algorithms based on some diagnostic features.

4.2: Objectives

1- Diabetes Prediction: The main objective is to utilize an existing ML model with high accuracy rate to classify whether a person is diabetic or not.
2 - Comparison of ML Classification Algorithms: This project will investigate and compare the performance and accuracy of different ML classification algorithms such as Decision tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Logistic Regression (LR) . The ML algorithm showing the most accuracy will be selected for further development.
3 - GUI Development: GUI is a part of this project in addition to building a ML model. It will be developed with user-friendly interface that enables the user to input certain diagnostic measurements so that the ML model can predict Type 2 Diabetes.

5 Methodology

5.1: Research Methodology and Approach

The research methodology chosen for this project is CRISP-DM Methodology since it is a data science project. [11]Cross-industry Standard Process for Data Mining (CRISP-DM) process begins with an iterative loop between business understanding and data understanding, passes through an iterative loop between data preparation and data modeling, and concludes with an evaluation phase that divides results between deployment and business understanding. Continuous data modeling, preparation, and evaluation result from the approach's development in a cyclic iterative loop.

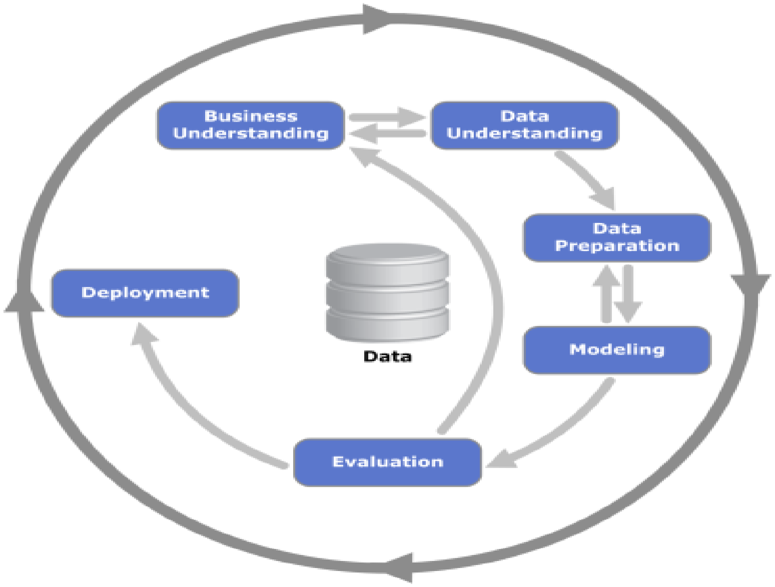


Figure-5-1: CRISP-DM Methodology (12)

There are six steps in CRISP-DM Methodology:

a) **Business Understanding:**
[12]It focuses on comprehending the requirements and goals of the project from a business standpoint. An analyst creates a preliminary plan and formulates this knowledge as a data mining problem.

b) **Data Understanding:**
[12]Beginning with basic data gathering, the analyst moves on to actions that will help him become acquainted with the data, identify data quality issues, and gain early insights into it. In this phase, the analyst may also find intriguing subsets to form hypotheses about hidden information.

c) **Data Preparation:**
[12]All tasks necessary to create the final dataset from the original raw data are included in the data preparation step.

d) **Modeling:**
[12]The analyst assesses, chooses, and uses suitable modeling approaches. Because many methods, like neural networks, have certain demands for the format of the data. This could lead back to data preparation. A loop may be created here.

e) **Evaluation:**
[12]The analyst selects and creates models that seem to be of high quality based on predetermined loss functions. After that, the analyst evaluates them to make sure the models can be applied to new data. The analyst then confirms that all important business challenges are adequately covered by the models. The choice of the champion model or models is the final outcome.

f) **Deployment:**
This usually entails integrating the model's code representation into an operating system. This also includes systems for rating or classifying newly discovered data as it becomes available. The procedure ought to apply the fresh

data to resolve the initial business issue. Crucially, all of the data preparation procedures that come before modeling must also be included in the code representation. This guarantees that the model will handle new raw data in the same way that it did when it was being developed.

I found CRISP-DM Methodology to be a suitable methodology for this project because it is neutral in terms of technology and problem. Anyone can use any software for analysis and apply it to whatever data mining challenge they desire. Regardless of the type of your data mining project, CRISP-DM will offer a framework with adequate structure.

The method of data collection used in this project is secondary data collection which is the process of gathering information from publicly accessible sources such as data published by people, groups, or organizations on open platforms, websites, or social media that can be accessed and used for research.

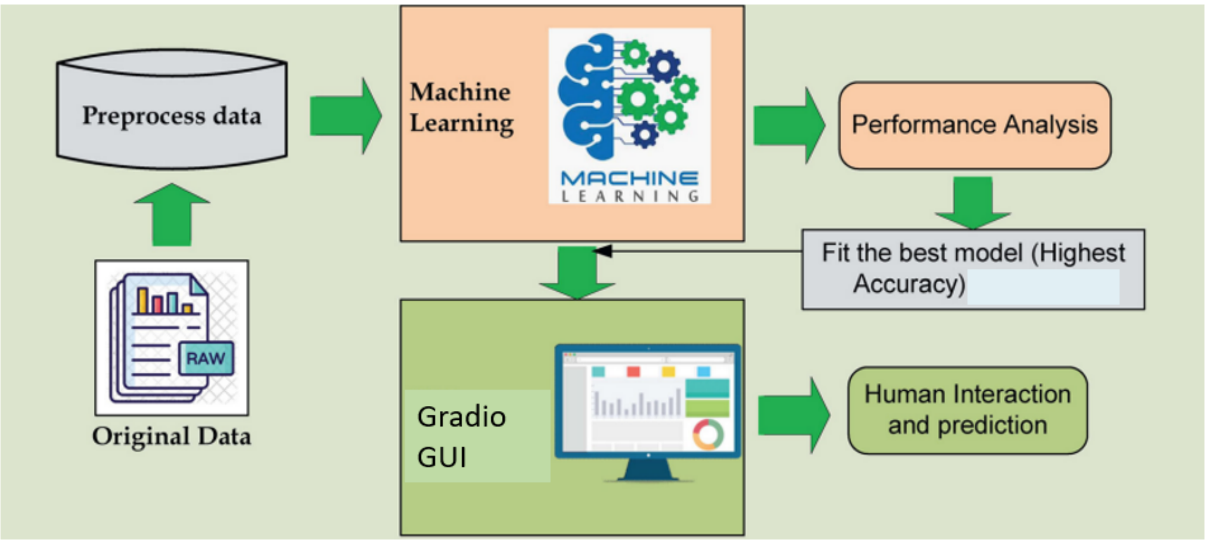


Figure 5-2: Project Methodology (Overview of the proposal, N. Ahmed, R. Ahammed, Md.M. Islam et al)

The research approach followed for this project is Quantitative research. It includes the collection and analysis of numerical data to objectively quantify and measure phenomena. [9]Numerical data is gathered, and statistical techniques are used for analysis in quantitative research. Producing objective, quantifiable, and numerically stated empirical data is the goal. Making predictions, finding patterns, and testing hypotheses are common uses for quantitative research. [10]Organized research equipment are typically used to collect the data. Higher sample sizes that are typical of the population are used to produce the results. Objective responses are sought for a well-defined study question. Non-textual forms such as tables, charts, and figures are frequently used to organize data, which are numerical and statistical in nature. Numerical data is gathered by researchers using instruments like computer software and surveys.

Information about the datasets:

Pima Indian Diabetes Dataset is one of the ideal datasets for evaluating machine learning algorithms for predicting diabetes (UCI Machine Learning Repository, 1998). This data set has been used by many researchers in predictive analyses [1], [2], [3], [4], [5]. The dataset consists of data used for diabetes research on women of Pima Indian heritage, aged 21 and over, living in Phoenix, the 5th largest city of the State of Arizona in the USA.

Description of Features:
Pregnancies: Number of times pregnant
Glucose: Plasma glucose concentration over 2 hours in an oral glucose tolerance test
BloodPressure: Diastolic blood pressure (mm Hg)
SkinThickness: Triceps skinfold thickness (mm)
Insulin: 2-Hour serum insulin (mu U/ml)
BMI: Body mass index (weight in kg/(height in m)^2)
Diabetes pedigree function - A function that scores likelihood of diabetes based on family history.
Age: Age in years
Outcome: Class variable (0: the person is not diabetic or 1: the person is diabetic)

Diabetes Prediction Dataset is a comprehensive dataset for predicting diabetes with medical and demographic information from patients, along with their diabetes status (positive or negative). The data includes features such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level.

Description of Features:
Gender - the biological sex of the individual, which can have an impact on their susceptibility to diabetes. There are three categories in it male,female and other.
Age - Age ranges from 0-80 in dataset. diabetes is more commonly diagnosed in older adults.
Hypertension - a medical condition in which the blood pressure in the arteries is persistently elevated. It has values a 0 or 1 where 0 indicates they don't have hypertension and for 1 it means they have hypertension.
Heart disease - a medical condition that is associated with an increased risk of developing diabetes. It has values a 0 or 1 where 0 indicates they don't have heart disease and for 1 it means they have heart disease.
Smoking history - It is a risk factor for diabetes and can exacerbate the complications associated with diabetes. There are 5 categories in the dataset i.e not current, former, No Info, current, never and ever.
Body Mass Index (BMI) - measurement of body fat based on weight and height. Higher BMI values are linked to a higher risk of diabetes. The range of BMI in the dataset is

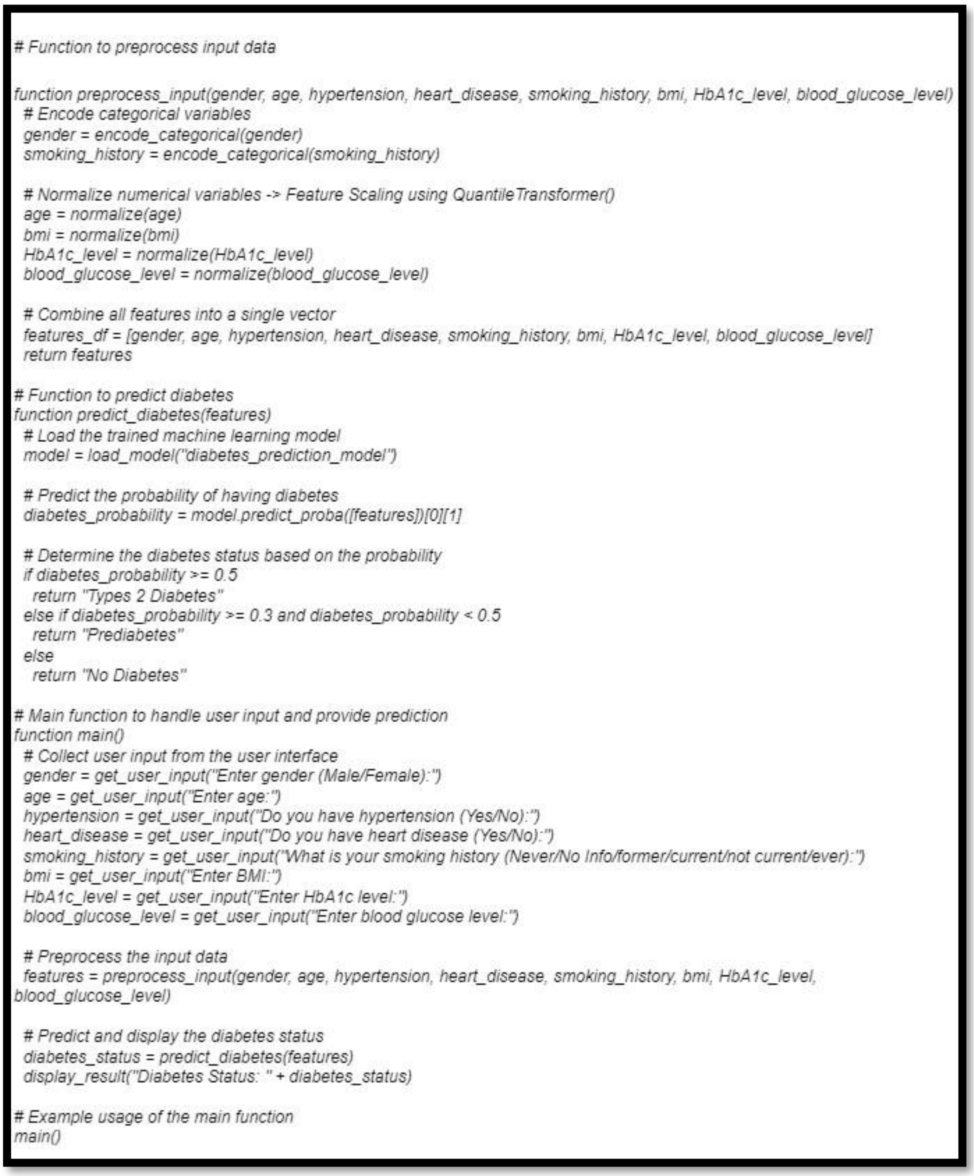
from 10.16 to 71.55. BMI less than 18.5 is underweight, 18.5-24.9 is normal, 25-29.9 is overweight, and 30 or more is obese.

Hemoglobin A1c (HbA1c) level - measurement of a person's average blood sugar level over the past 2-3 months. Higher levels indicate a greater risk of developing diabetes. Mostly more than 6.5% of HbA1c Level indicates diabetes.

Blood glucose level - the amount of glucose in the bloodstream at a given time. High blood glucose levels are a key indicator of diabetes.

Diabetes - the target variable being predicted, with values of 1 indicating the presence of diabetes and 0 indicating the absence of diabetes.

Pseudocode for building ML model to predict T2D in a person.



```
# Function to preprocess input data

function preprocess_input(gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, blood_glucose_level)
  # Encode categorical variables
  gender = encode_categorical(gender)
  smoking_history = encode_categorical(smoking_history)

  # Normalize numerical variables -> Feature Scaling using QuantileTransformer()
  age = normalize(age)
  bmi = normalize(bmi)
  HbA1c_level = normalize(HbA1c_level)
  blood_glucose_level = normalize(blood_glucose_level)

  # Combine all features into a single vector
  features_df = [gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level, blood_glucose_level]
  return features

# Function to predict diabetes
function predict_diabetes(features)
  # Load the trained machine learning model
  model = load_model("diabetes_prediction_model")

  # Predict the probability of having diabetes
  diabetes_probability = model.predict_proba([features])[0][1]

  # Determine the diabetes status based on the probability
  if diabetes_probability >= 0.5
    return "Types 2 Diabetes"
  else if diabetes_probability >= 0.3 and diabetes_probability < 0.5
    return "Prediabetes"
  else
    return "No Diabetes"

# Main function to handle user input and provide prediction
function main()
  # Collect user input from the user interface
  gender = get_user_input("Enter gender (Male/Female):")
  age = get_user_input("Enter age:")
  hypertension = get_user_input("Do you have hypertension (Yes/No):")
  heart_disease = get_user_input("Do you have heart disease (Yes/No):")
  smoking_history = get_user_input("What is your smoking history (Never/No Info/former/current/not current/ever):")
  bmi = get_user_input("Enter BMI:")
  HbA1c_level = get_user_input("Enter HbA1c level:")
  blood_glucose_level = get_user_input("Enter blood glucose level:")

  # Preprocess the input data
  features = preprocess_input(gender, age, hypertension, heart_disease, smoking_history, bmi, HbA1c_level,
blood_glucose_level)

  # Predict and display the diabetes status
  diabetes_status = predict_diabetes(features)
  display_result("Diabetes Status: " + diabetes_status)

# Example usage of the main function
main()
```

Figure 5-3: Pseudocode

5.3: System Requirements

5.3.1: Software Requirements:

- Python
- Google Colaboratory
- Gradio Interface

5.3.2: Hardware Requirements:

- Processor: Intel Core i7
- RAM: 16GB
- OS: Windows 10

5.3.3: Python Libraries:

- Pandas – used to read the dataset in csv format and create dataframe.
- NumPy – used to perform a wide variety of mathematical operations on arrays
- Matplotlib and Seaborn – used for data visualization
- Sklearn – used for feature scaling, importing ML models, evaluate performance matrix, etc.

6 Design and Implementation

6.1: Machine Learning Process

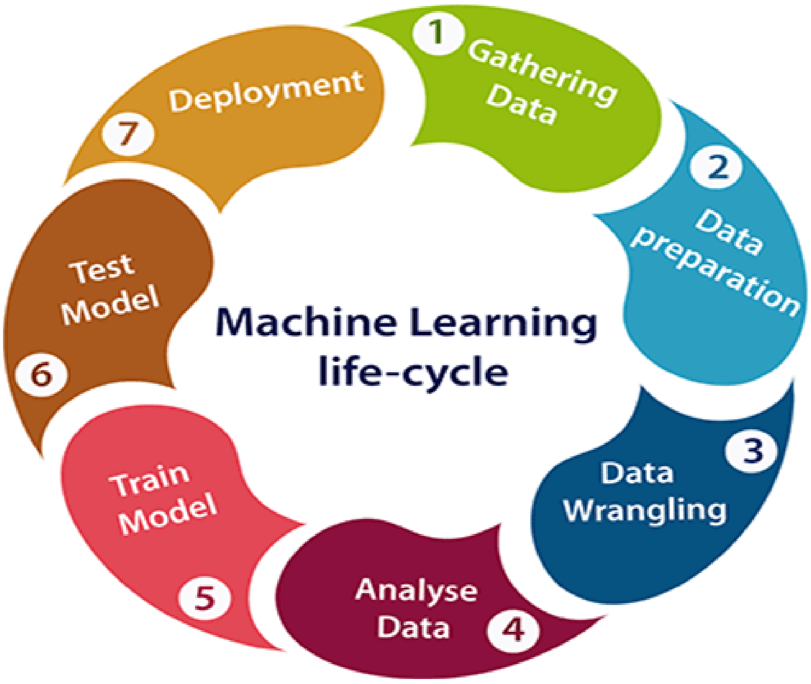


Figure 6-1: Machine Learning Life Cycle [13]

“Pima Indian Diabetes Dataset” and "Diabetes Prediction Dataset" is referred as Dataset 1 and Dataset 2 for simplicity.

The following list of actions are taken to accomplish this project's goal:

A. Data Analysis & Machine Learning

i. **Data Collection:**

“Pima Indian Diabetes Dataset” and "Diabetes Prediction Dataset" are public datasets from Kaggle which will be utilized for this project. The CSV files of both the datasets are downloaded from Kaggle and uploaded into Google Drive. A new Google Colab notebook is created, necessary python libraries are imported, and Google Drive is mounted for data analysis and machine learning to be done on both datasets.

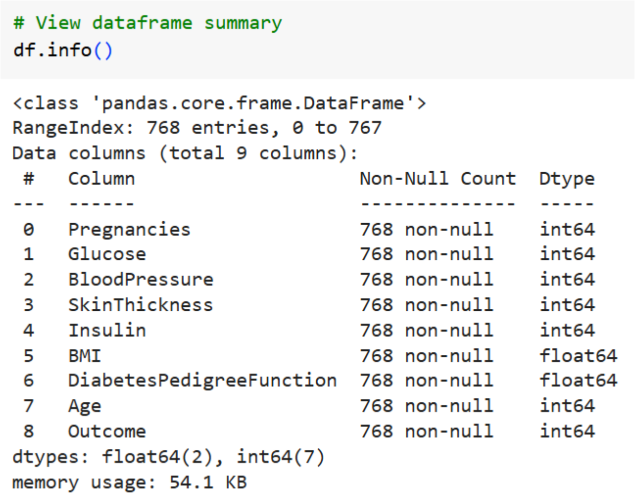
ii. **Data Preprocessing and Data Cleaning:**

Each dataset is imported to Colab notebook and a dataframe df is created. Statistical analysis is done for both datasets using the following preprocessing techniques:

df.head() - view the top 5 rows of df

df.shape() – view the dimensions of dataframe, i.e., no. of rows and columns. By There are 9 columns and 768 rows in Dataset 1 dataframe and 9 columns and 100000 rows in Dataset 2 dataframe.

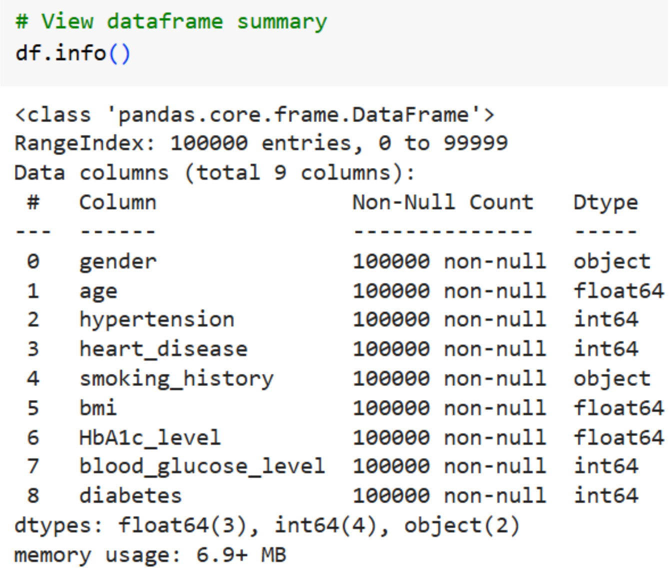
df.info() – view summary of df



```
# View dataframe summary
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Figure 6-3: Summary of Dataset 1

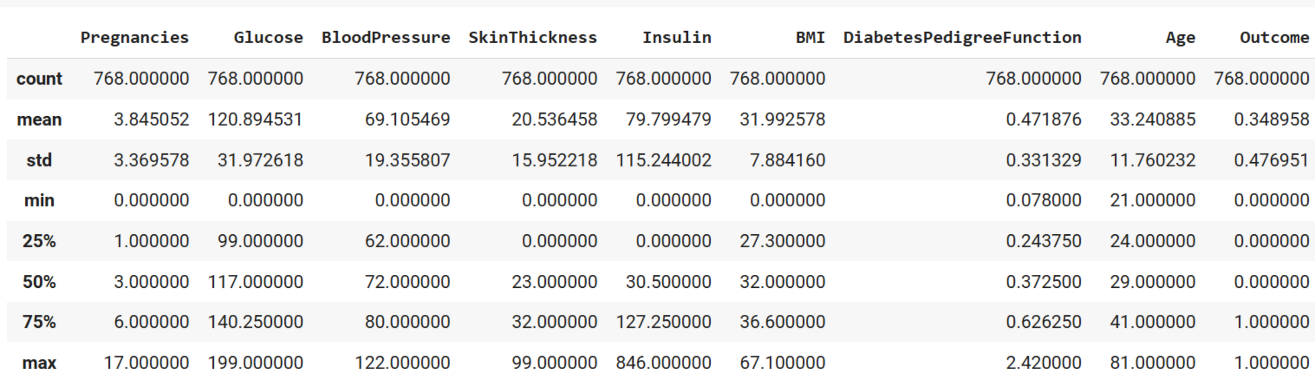


```
# View dataframe summary
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 9 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   gender               100000 non-null  object
 1   age                  100000 non-null  float64
 2   hypertension         100000 non-null  int64
 3   heart_disease        100000 non-null  int64
 4   smoking_history      100000 non-null  object
 5   bmi                  100000 non-null  float64
 6   HbA1c_level          100000 non-null  float64
 7   blood_glucose_level  100000 non-null  int64
 8   diabetes             100000 non-null  int64
dtypes: float64(3), int64(4), object(2)
memory usage: 6.9+ MB
```

Figure 6-2: Summary of Dataset 2

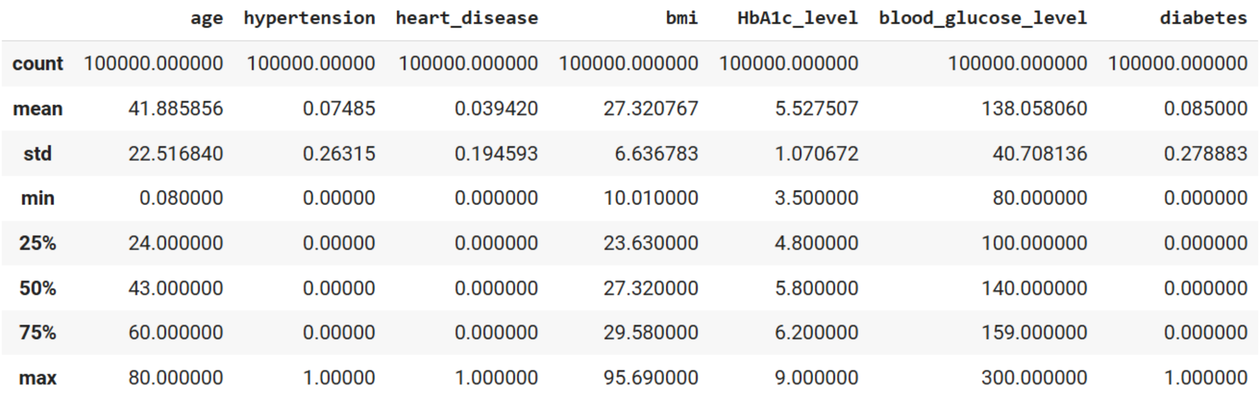
df.describe() - View descriptive statistics



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

Figure 6-4: Descriptive statistics for Dataset 1

For Dataset 1, It can be noticed that Pregnancies appear in a realistic range from 0 to 17, DiabetesPedigreeFunction is in a realistic range of 0.08 to 2.42, Age has a realistic range from 21 to 81. The Outcome, in the target variable, 0 represents healthy people, and 1 represents those with diabetes. Some other attributes in the data (Glucose, BloodPressure, SkinThickness, Insulin, BMI) include the value 0, which is not possible in practice. In this case, the impossible 0 values need to be corrected.



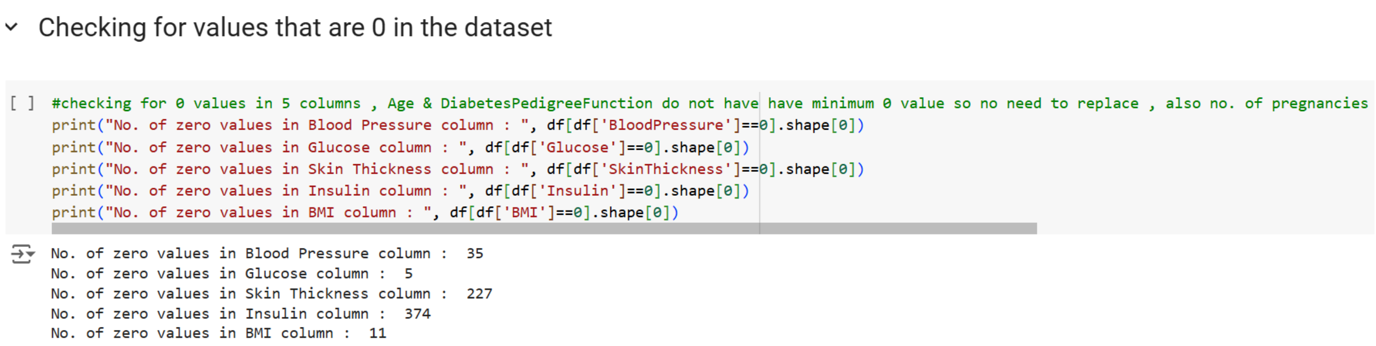
	age	hypertension	heart_disease	bmi	HbA1c_level	blood_glucose_level	diabetes
count	100000.000000	100000.00000	100000.000000	100000.000000	100000.000000	100000.000000	100000.000000
mean	41.885856	0.07485	0.039420	27.320767	5.527507	138.058060	0.085000
std	22.516840	0.26315	0.194593	6.636783	1.070672	40.708136	0.278883
min	0.080000	0.00000	0.000000	10.010000	3.500000	80.000000	0.000000
25%	24.000000	0.00000	0.000000	23.630000	4.800000	100.000000	0.000000
50%	43.000000	0.00000	0.000000	27.320000	5.800000	140.000000	0.000000
75%	60.000000	0.00000	0.000000	29.580000	6.200000	159.000000	0.000000
max	80.000000	1.00000	1.000000	95.690000	9.000000	300.000000	1.000000

Figure 6-5: Descriptive statistics for Dataset 2

For dataset 2, values for all columns appears to be within the realistic range and there are no impossible values.

df.isnull().values.any() - returns True when there is at least one missing value occurring in the data. It was found that there were no null column values in Dataset 2 but there are many zero column values in Dataset 1.

df.duplicated().sum() – checking for duplicate values. The output showed that there are no duplicate values in Dataset 1 but there are about 3854 duplicate values in Dataset 2. So, df.drop_duplicates() was used to drop those duplicate values.



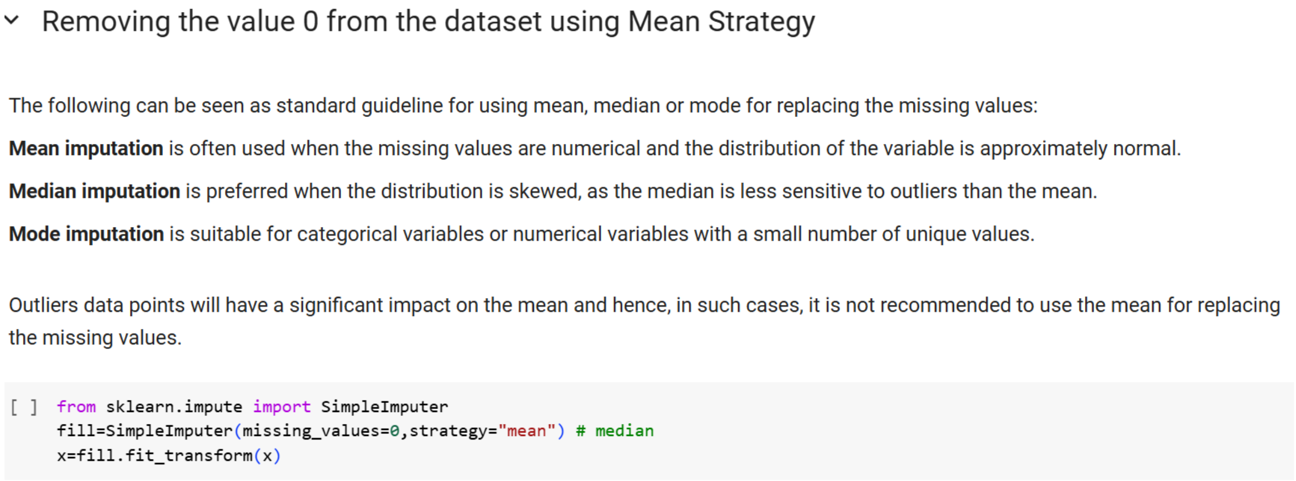
Checking for values that are 0 in the dataset

```
[ ] #checking for 0 values in 5 columns , Age & DiabetesPedigreeFunction do not have have minimum 0 value so no need to replace , also no. of pregnancies
    print("No. of zero values in Blood Pressure column : ", df[df['BloodPressure']==0].shape[0])
    print("No. of zero values in Glucose column : ", df[df['Glucose']==0].shape[0])
    print("No. of zero values in Skin Thickness column : ", df[df['SkinThickness']==0].shape[0])
    print("No. of zero values in Insulin column : ", df[df['Insulin']==0].shape[0])
    print("No. of zero values in BMI column : ", df[df['BMI']==0].shape[0])
```

```
No. of zero values in Blood Pressure column :  35
No. of zero values in Glucose column :  5
No. of zero values in Skin Thickness column :  227
No. of zero values in Insulin column :  374
No. of zero values in BMI column :  11
```

Figure 6-6: Zero values checking for Dataset 1

It is shown that there are many zero values in the columns of Dataset 1. This issue can be resolved by using Mean Strategy.



Removing the value 0 from the dataset using Mean Strategy

The following can be seen as standard guideline for using mean, median or mode for replacing the missing values:

Mean imputation is often used when the missing values are numerical and the distribution of the variable is approximately normal.

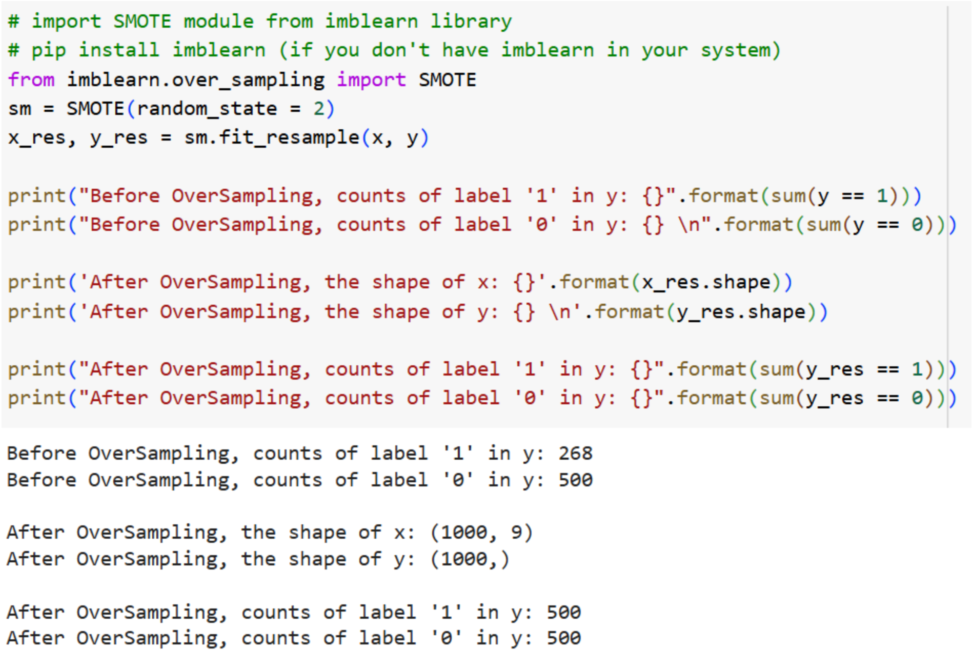
Median imputation is preferred when the distribution is skewed, as the median is less sensitive to outliers than the mean.

Mode imputation is suitable for categorical variables or numerical variables with a small number of unique values.

Outliers data points will have a significant impact on the mean and hence, in such cases, it is not recommended to use the mean for replacing the missing values.

```
[ ] from sklearn.impute import SimpleImputer
    fill=SimpleImputer(missing_values=0,strategy="mean") # median
    x=fill.fit_transform(x)
```

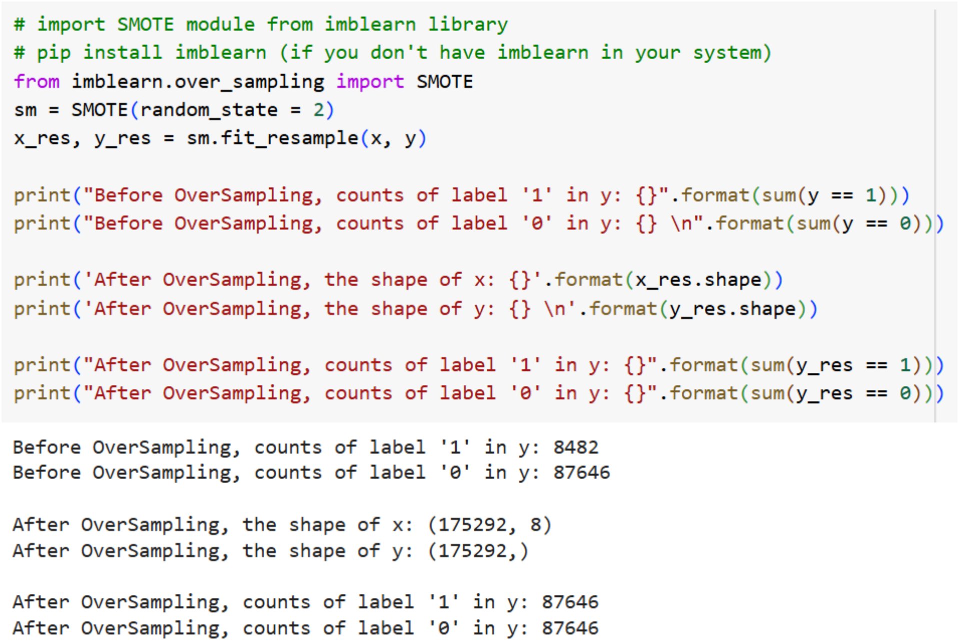
Figure 6-7: Remove zero values in Dataset 1



```
# import SMOTE module from imblearn library
# pip install imblearn (if you don't have imblearn in your system)
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
x_res, y_res = sm.fit_resample(x, y)

print("Before OverSampling, counts of label '1' in y: {}".format(sum(y == 1)))
print("Before OverSampling, counts of label '0' in y: {} \n".format(sum(y == 0)))

print('After OverSampling, the shape of x: {}'.format(x_res.shape))
print('After OverSampling, the shape of y: {} \n'.format(y_res.shape))

print("After OverSampling, counts of label '1' in y: {}".format(sum(y_res == 1)))
print("After OverSampling, counts of label '0' in y: {}".format(sum(y_res == 0)))
```

```
Before OverSampling, counts of label '1' in y: 268
Before OverSampling, counts of label '0' in y: 500

After OverSampling, the shape of x: (1000, 9)
After OverSampling, the shape of y: (1000,)

After OverSampling, counts of label '1' in y: 500
After OverSampling, counts of label '0' in y: 500
```

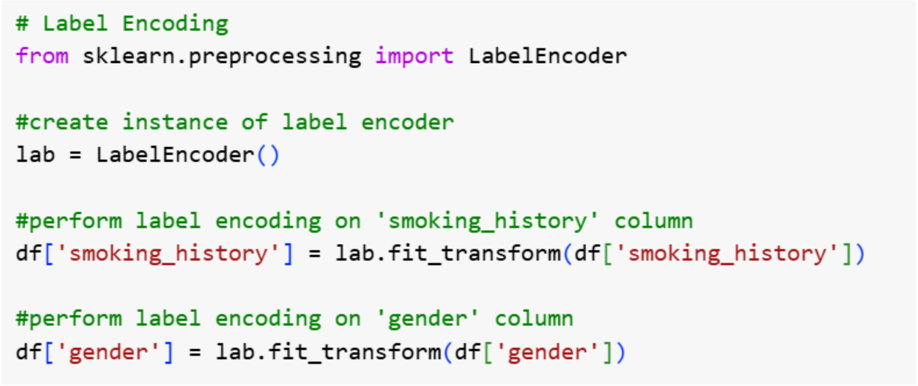
Figure 6-8: SMOTE for Dataset 1



```
# import SMOTE module from imblearn library
# pip install imblearn (if you don't have imblearn in your system)
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
x_res, y_res = sm.fit_resample(x, y)

print("Before OverSampling, counts of label '1' in y: {}".format(sum(y == 1)))
print("Before OverSampling, counts of label '0' in y: {} \n".format(sum(y == 0)))

print('After OverSampling, the shape of x: {}'.format(x_res.shape))
print('After OverSampling, the shape of y: {} \n'.format(y_res.shape))

print("After OverSampling, counts of label '1' in y: {}".format(sum(y_res == 1)))
print("After OverSampling, counts of label '0' in y: {}".format(sum(y_res == 0)))
```

```
Before OverSampling, counts of label '1' in y: 8482
Before OverSampling, counts of label '0' in y: 87646

After OverSampling, the shape of x: (175292, 8)
After OverSampling, the shape of y: (175292,)

After OverSampling, counts of label '1' in y: 87646
After OverSampling, counts of label '0' in y: 87646
```

Figure 6-9: SMOTE for Dataset 2



```
# Label Encoding
from sklearn.preprocessing import LabelEncoder

#create instance of label encoder
lab = LabelEncoder()

#perform label encoding on 'smoking_history' column
df['smoking_history'] = lab.fit_transform(df['smoking_history'])

#perform label encoding on 'gender' column
df['gender'] = lab.fit_transform(df['gender'])
```

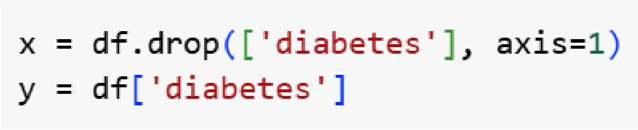
Figure 6-10: Label Encoding Dataset 2

Dataset 2 contains columns having categorical values like ‘smoking_history’ and ‘gender’. So, Label Encoding is used to convert categorical values into numerical values and it is labeled.

iii. Data Visualization

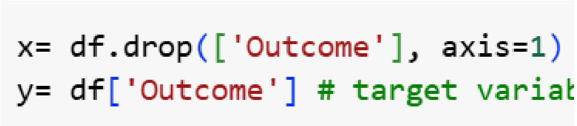
Data visualization tools were used to visualize the data and find out interesting insights and hidden patterns in the dataset like Boxplot, Count plot, Pie Chart, Bar plot, hist plot, pair plot, Heatmap used for feature selection and finding the correlation between the features in the dataset and figure out which feature has more importance than other features in the dataset.

iv. Segregating the dataset into features(x) and target variable(y) and splitting x and y in the ratio 70% training set and 30% test set.



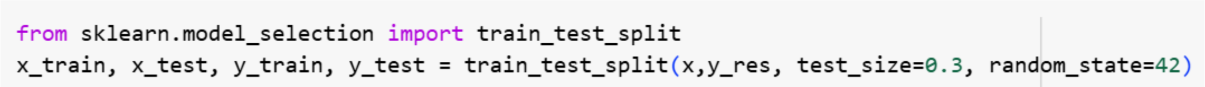
```
x = df.drop(['diabetes'], axis=1)
y = df['diabetes']
```

Figure 6-12:x,y Dataset 2



```
x= df.drop(['Outcome'], axis=1)
y= df['Outcome'] # target variat
```

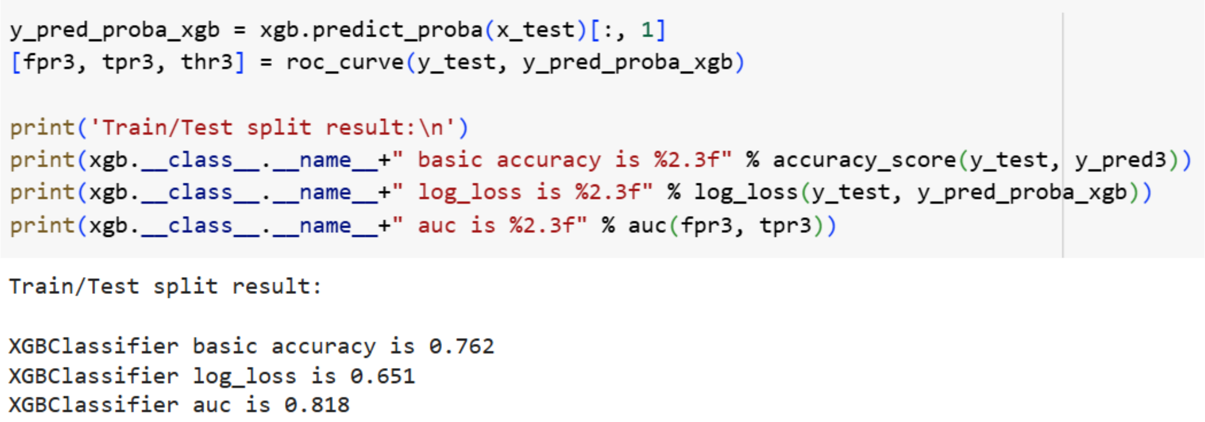
Figure 6-11: x,y Dataset 1



```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y_res, test_size=0.3, random_state=42)
```

Figure 6-13: Train, test, split

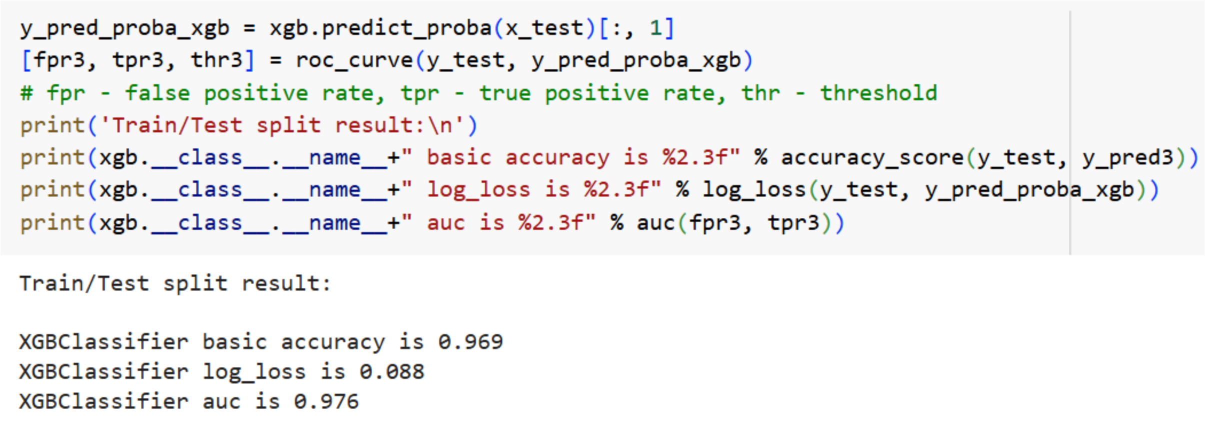
v. Building predictive models using ML algorithms like XGBoost, Random Forest, Decision Tree, Logistic Regression to train the dataset to make accurate predictions for both Datasets 1 and 2.



```
y_pred_proba_xgb = xgb.predict_proba(x_test)[:, 1]
[fpr3, tpr3, thr3] = roc_curve(y_test, y_pred_proba_xgb)

print('Train/Test split result:\n')
print(xgb.__class__.__name__+" basic accuracy is %2.3f" % accuracy_score(y_test, y_pred3))
print(xgb.__class__.__name__+" log_loss is %2.3f" % log_loss(y_test, y_pred_proba_xgb))
print(xgb.__class__.__name__+" auc is %2.3f" % auc(fpr3, tpr3))
```

```
Train/Test split result:

XGBClassifier basic accuracy is 0.762
XGBClassifier log_loss is 0.651
XGBClassifier auc is 0.818
```

Figure 6-14: XGB Dataset 1



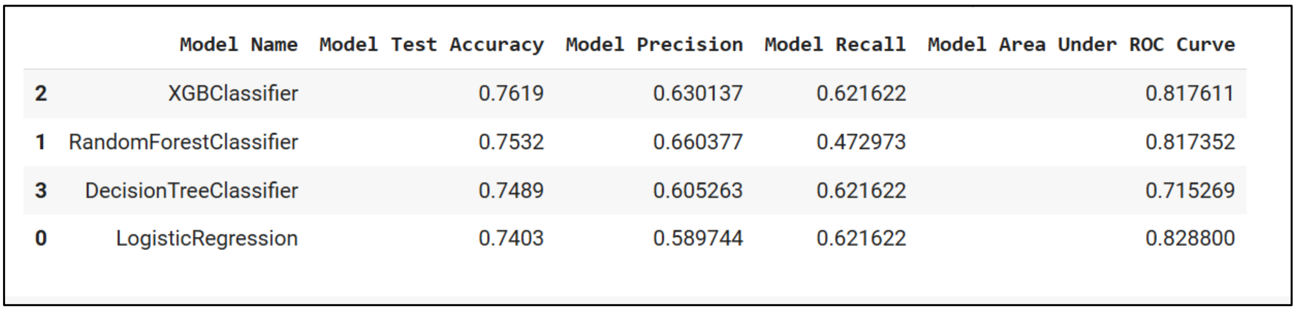
```
y_pred_proba_xgb = xgb.predict_proba(x_test)[:, 1]
[fpr3, tpr3, thr3] = roc_curve(y_test, y_pred_proba_xgb)
# fpr - false positive rate, tpr - true positive rate, thr - threshold
print('Train/Test split result:\n')
print(xgb.__class__.__name__+" basic accuracy is %2.3f" % accuracy_score(y_test, y_pred3))
print(xgb.__class__.__name__+" log_loss is %2.3f" % log_loss(y_test, y_pred_proba_xgb))
print(xgb.__class__.__name__+" auc is %2.3f" % auc(fpr3, tpr3))
```

```
Train/Test split result:

XGBClassifier basic accuracy is 0.969
XGBClassifier log_loss is 0.088
XGBClassifier auc is 0.976
```

Figure 6-15: XGB Dataset 2

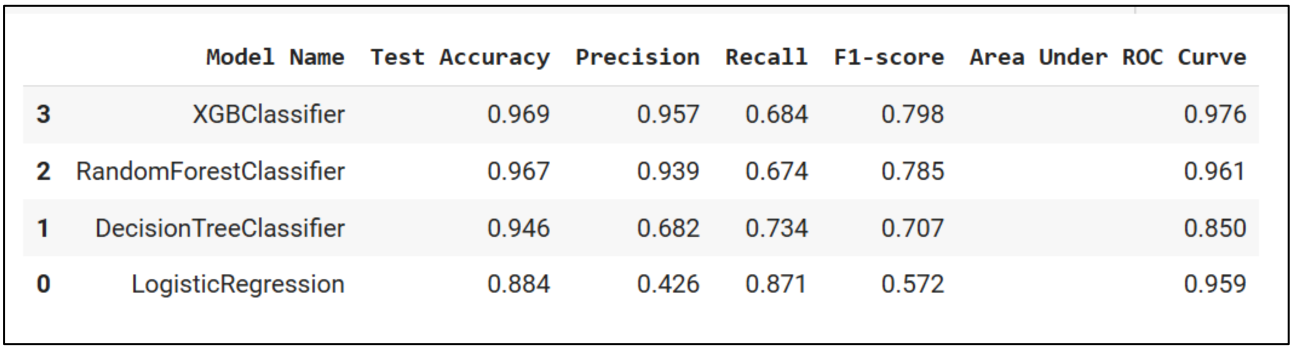
vi. Evaluating the ML models using performance matrix of each model.
The performance of all the predictive models was evaluated on the basis of performance matrix like accuracy, precision, recall, f1-score, area under ROC curve, and confusion matrix for Dataset 1 and 2.

Accuracy is the ratio of correctly predicted observations to the total observations. **Precision** expresses the ratio of correctly detected Positive classes to all positives. Recall expresses the ratio of correctly detected Positive classes to true positives. The **F1-score** is the harmonic mean of sensitivity and precision. **Area under the ROC Curve (AUC)** value measures the accuracy of a diagnostic test. It is calculated according to the area under the Receiver Operating Characteristic (ROC) curve. **Confusion Matrix** is used to measure the performance of the classification model. It has four different outcomes: TP, TN, FP, FN.



	Model Name	Model Test Accuracy	Model Precision	Model Recall	Model Area Under ROC Curve
2	XGBClassifier	0.7619	0.630137	0.621622	0.817611
1	RandomForestClassifier	0.7532	0.660377	0.472973	0.817352
3	DecisionTreeClassifier	0.7489	0.605263	0.621622	0.715269
0	LogisticRegression	0.7403	0.589744	0.621622	0.828800

Figure 6-16: Evaluation Matrix Dataset 1



	Model Name	Test Accuracy	Precision	Recall	F1-score	Area Under ROC Curve
3	XGBClassifier	0.969	0.957	0.684	0.798	0.976
2	RandomForestClassifier	0.967	0.939	0.674	0.785	0.961
1	DecisionTreeClassifier	0.946	0.682	0.734	0.707	0.850
0	LogisticRegression	0.884	0.426	0.871	0.572	0.959

Figure 6-17: Evaluation Matrix Dataset 2

It can be inferred from the results that XGBoost model has highest accuracy rate and performs well compared to other model. This result was observed in Dataset 1 and 2.

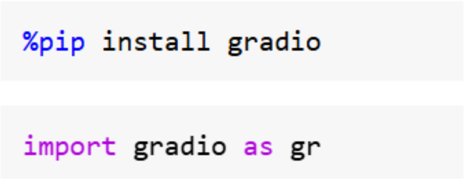
6.2: GUI Development

1. XGBoost model with the highest accuracy rate of 97% and Dataset 2 is chosen for further development.

2. Using Google Colab and Gradio interface, create a user-friendly GUI and deploy the most accurate machine learning model.

[14]Gradio is an open-source Python tool that allows you to quickly create a demo or web application for any machine learning model, API, or any dynamically defined Python function. Gradio's built-in sharing features allow you to quickly distribute a link to your demo or web application. There is no requirement for prior JavaScript, CSS, or web hosting knowledge.

[14]It is advised to use pip command , which is included by default in Python, to install Gradio. The terminal or command prompt can be used to run the code. Gradio can be used anywhere Python is written, including in Google Colab, Jupyter notebooks, and code editors of your choice.



```
%pip install gradio
```

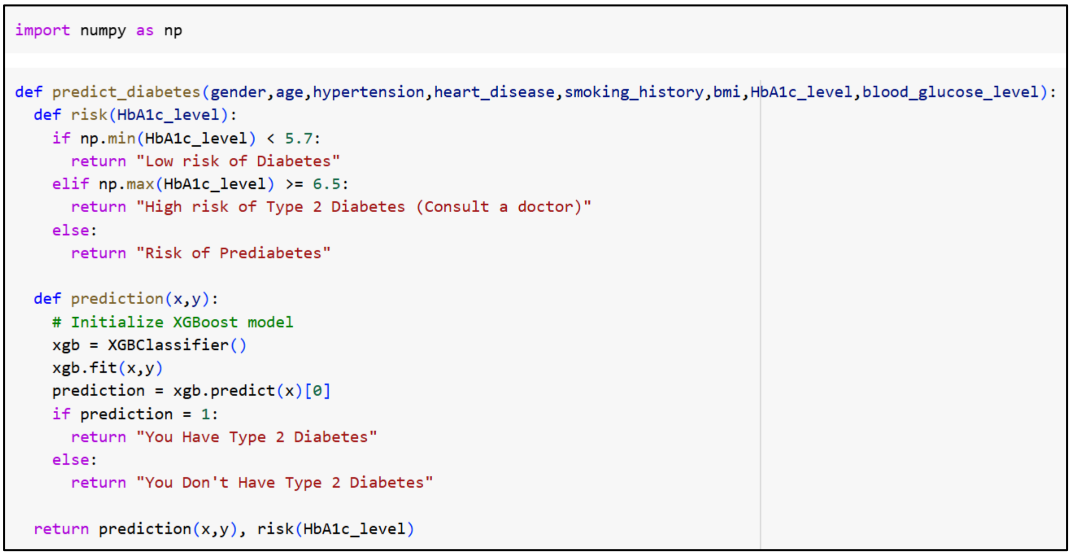
```
import gradio as gr
```

Figure 6-18: Install and import Gradio Interface

Comprehending the Interface Class

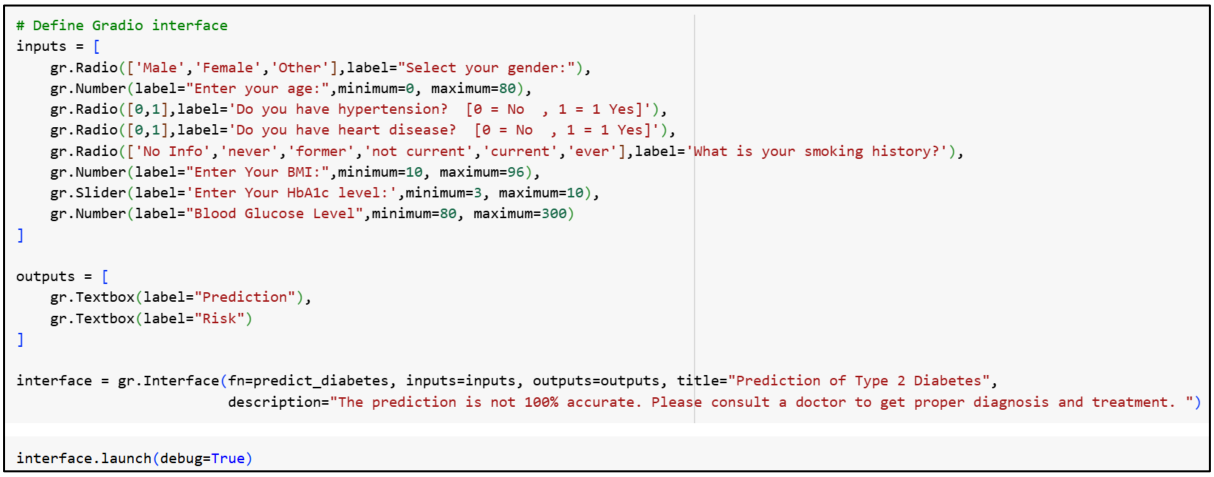
[14] Machine learning models that receive one or more inputs and return one or more outputs can be demonstrated using the Interface class. There are three main arguments for the Interface class:
fn: the function to wrap a user interface (UI).
inputs: The Gradio component or components to be used as an input. The quantity of components and arguments in your function should be equal.
outputs: The Gradio component or components to be used in the output. The quantity of components and return values from your function should be uniform.

[14]The input and output arguments accept one or more Gradio components. Gradio comes with over 30 built-in components designed for machine learning applications, including gr.Textbox(), gr.Image(), and gr.HTML(). Because of its flexibility, the Interface class is an extremely valuable tool for creating demos.



```
import numpy as np

def predict_diabetes(gender,age,hypertension,heart_disease,smoking_history,bmi,HbA1c_level,blood_glucose_level):
  def risk(HbA1c_level):
    if np.min(HbA1c_level) < 5.7:
      return "Low risk of Diabetes"
    elif np.max(HbA1c_level) >= 6.5:
      return "High risk of Type 2 Diabetes (Consult a doctor)"
    else:
      return "Risk of Prediabetes"

  def prediction(x,y):
    # Initialize XGBoost model
    xgb = XGBClassifier()
    xgb.fit(x,y)
    prediction = xgb.predict(x)[0]
    if prediction = 1:
      return "You Have Type 2 Diabetes"
    else:
      return "You Don't Have Type 2 Diabetes"

  return prediction(x,y), risk(HbA1c_level)
```

Figure 6-19: Gradio Function



```
# Define Gradio interface
inputs = [
    gr.Radio(['Male','Female','Other'],label="Select your gender:"),
    gr.Number(label="Enter your age:",minimum=0, maximum=80),
    gr.Radio([0,1],label='Do you have hypertension?  [0 = No  , 1 = 1 Yes]'),
    gr.Radio([0,1],label='Do you have heart disease?  [0 = No  , 1 = 1 Yes]'),
    gr.Radio(['No Info','never','former','not current','current','ever'],label='What is your smoking history?'),
    gr.Number(label="Enter Your BMI:",minimum=10, maximum=96),
    gr.Slider(label='Enter Your HbA1c level:',minimum=3, maximum=10),
    gr.Number(label="Blood Glucose Level",minimum=80, maximum=300)
]

outputs = [
    gr.Textbox(label="Prediction"),
    gr.Textbox(label="Risk")
]

interface = gr.Interface(fn=predict_diabetes, inputs=inputs, outputs=outputs, title="Prediction of Type 2 Diabetes",
                         description="The prediction is not 100% accurate. Please consult a doctor to get proper diagnosis and treatment. ")

interface.launch(debug=True)
```

Figure 6-20: Gradio Design

3. After running the code, GUI will appear. The GUI was function inside the Colab notebook, or it can be opened in a web browser new tab or window by clicking on the public URL or it can be deployed in Hugging Face Spaces permanently. Now, users can enter medical data to determine if they have Type 2 diabetes or not.

7 Result and Analysis

7.1: Interpretation of Data Visualization

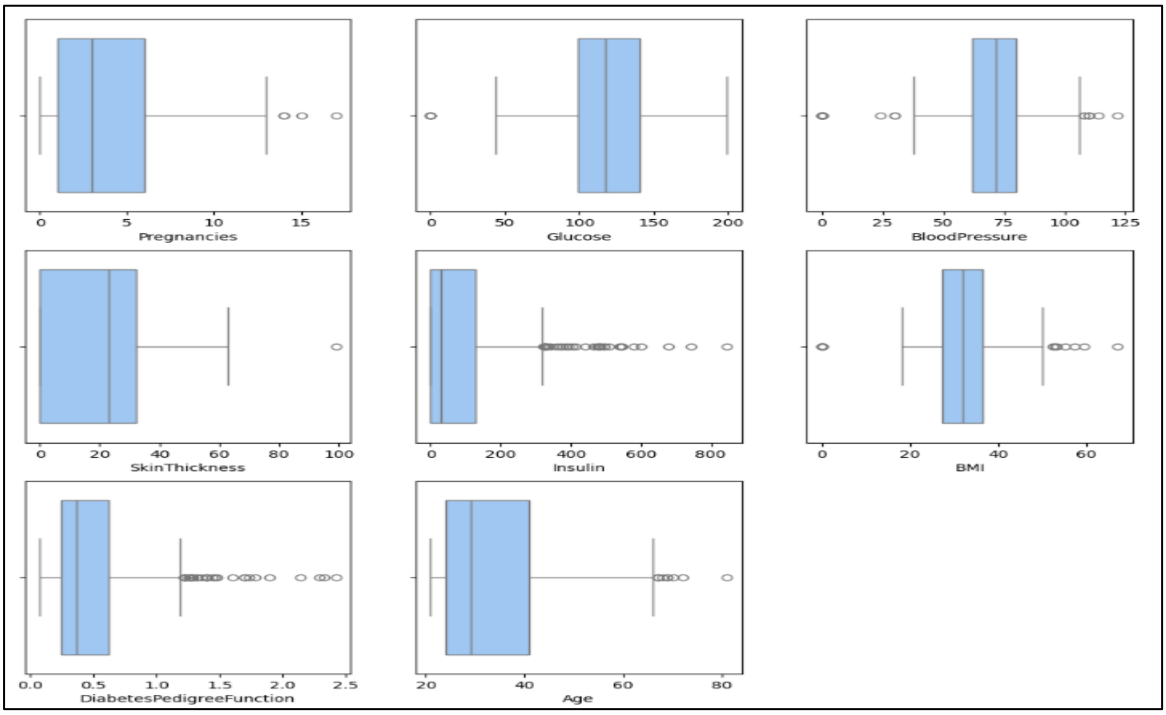


Figure 7-1: Boxplot for Dataset 1

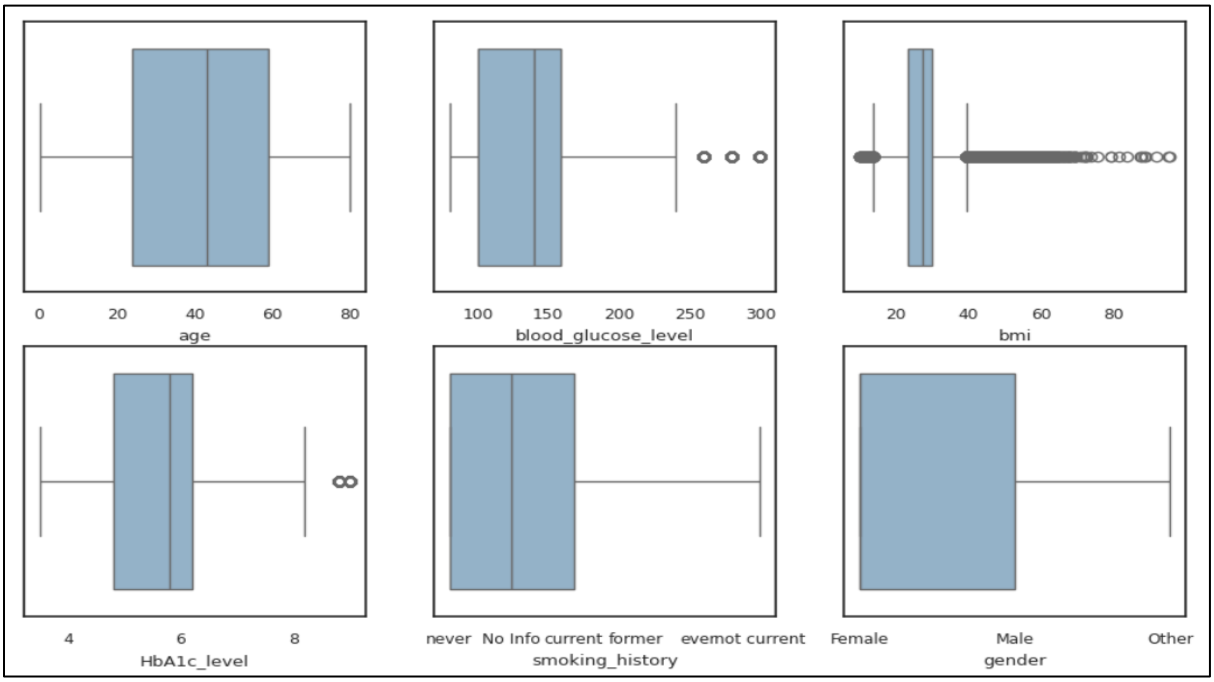


Figure 7-2: Boxplot for Dataset 2

Box plots were plotted to identify outliers in both the datasets. Several outliers were found in Dataset 1 which were not realistic and impossible values. This issue can be mitigated using QuantileTransformer() to feature scale the dataframe columns. This method transforms the features to follow a uniform or a normal distribution. Therefore, for a given feature, this transformation tends to spread out the most frequent values. It also reduces the impact of (marginal) outliers and is a robust preprocessing scheme. After applying feature scaling there were no outliers in Dataset 1. The outliers in Dataset 2 are realistic values are not impossible values. Eg. BMI.

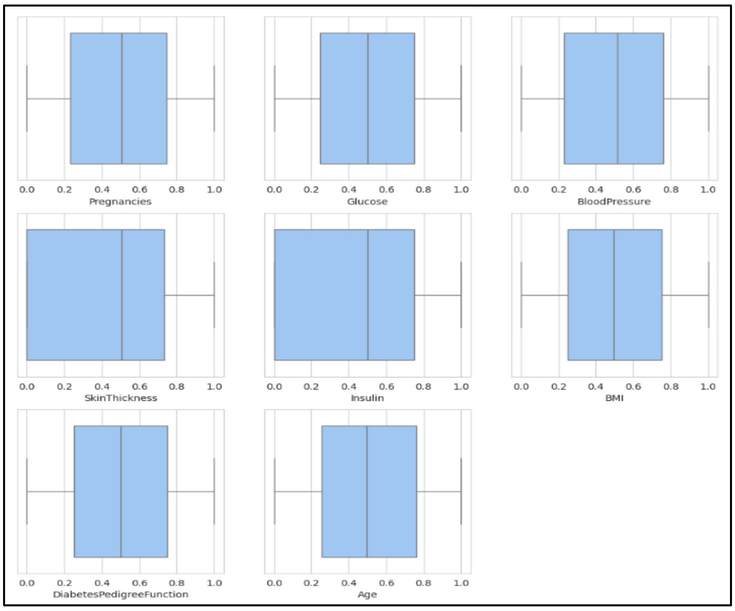
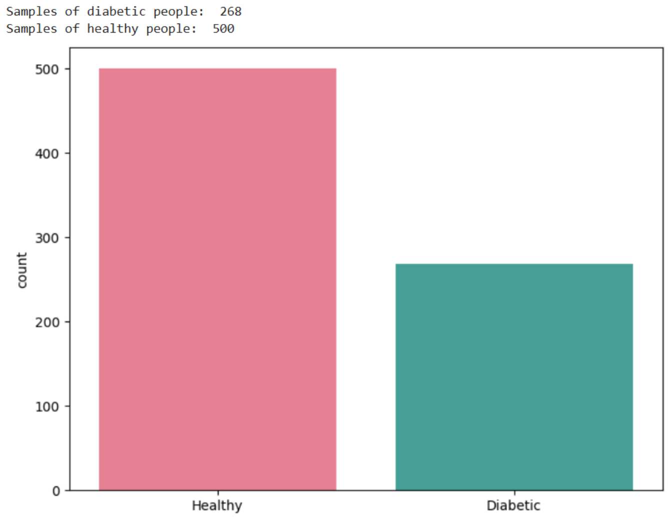


Figure 7-3: Boxplot for Dataset 1 Quantile Transformer

Countplot can be plotted of target variable (Outcome for Dataset 1 and Diabetes for Dataset 2) to check if there is class imbalance. Class imbalance refers to a situation in a dataset where the distribution of classes or categories is highly uneven. In such cases, one or more classes have significantly fewer instances compared to others.

Before Synthetic Minority Over-sampling Technique (SMOTE)



```
Samples of diabetic people:  268
Samples of healthy people:  500
```

Figure 7-5: Target variable distribution for Dataset 1

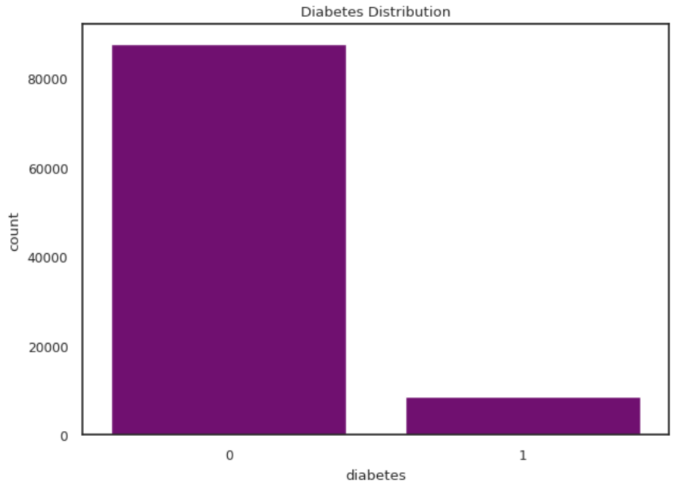
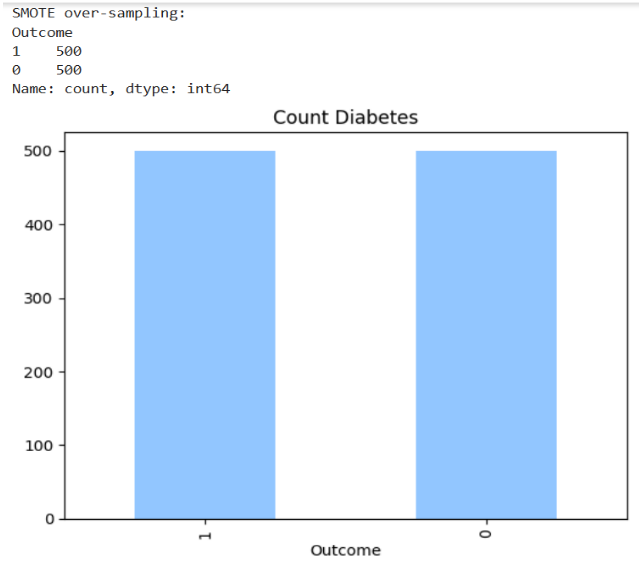


Figure 7-4:Target variable distribution for Dataset 2

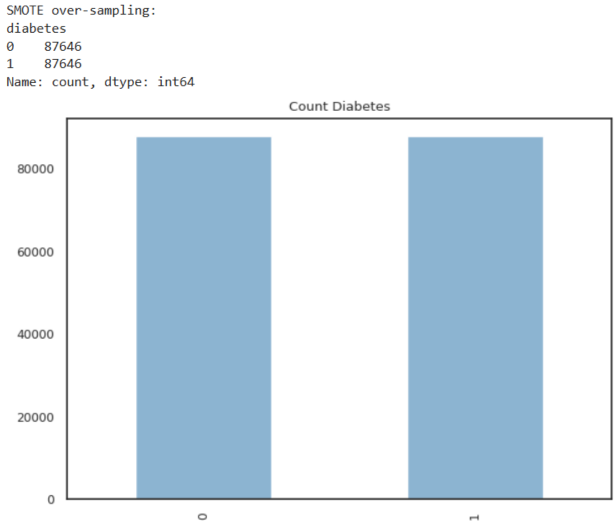
It is found that instances for 0 (No Diabetes) is more than 1 (Diabetes) for both the datasets 1 and 2. The majority class is 0 (No Diabetes) and the minority class is than 1 (Diabetes). To balance it, oversampling should be done for the minority class. [16]This issue is mitigated by taking advantage of SMOTE Oversampling technique.

After Synthetic Minority Over-sampling Technique (SMOTE)



```
SMOTE over-sampling:
Outcome
1    500
0    500
Name: count, dtype: int64
```

Figure 7-6: SMOTE Target variable distribution for Dataset 1



```
SMOTE over-sampling:
diabetes
0    87646
1    87646
Name: count, dtype: int64
```

Figure 7-7: SMOTE Target variable distribution for Dataset 2

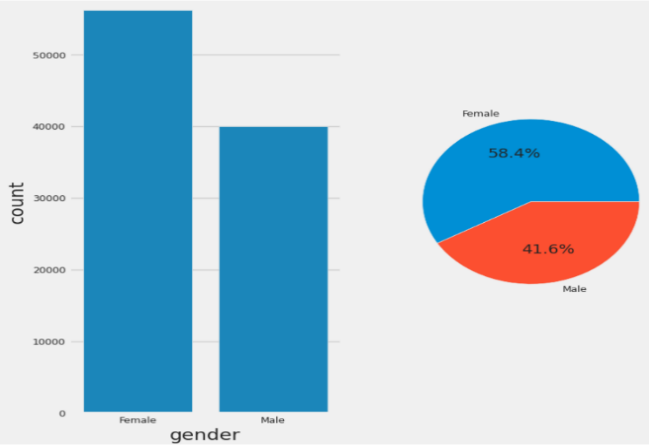


Figure 7-9: Gender Distribution Dataset 2

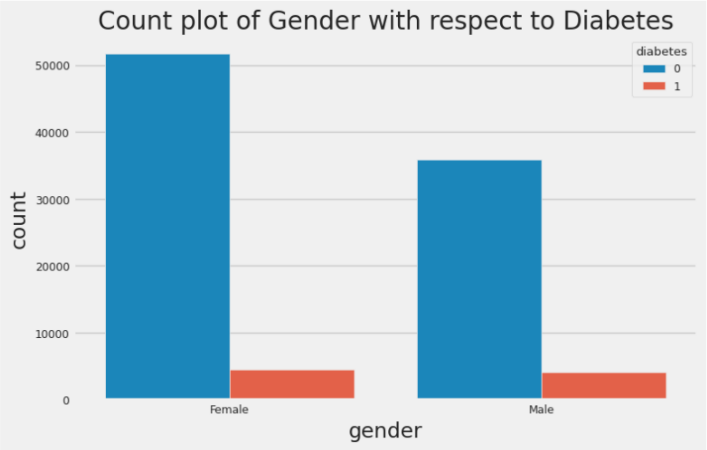
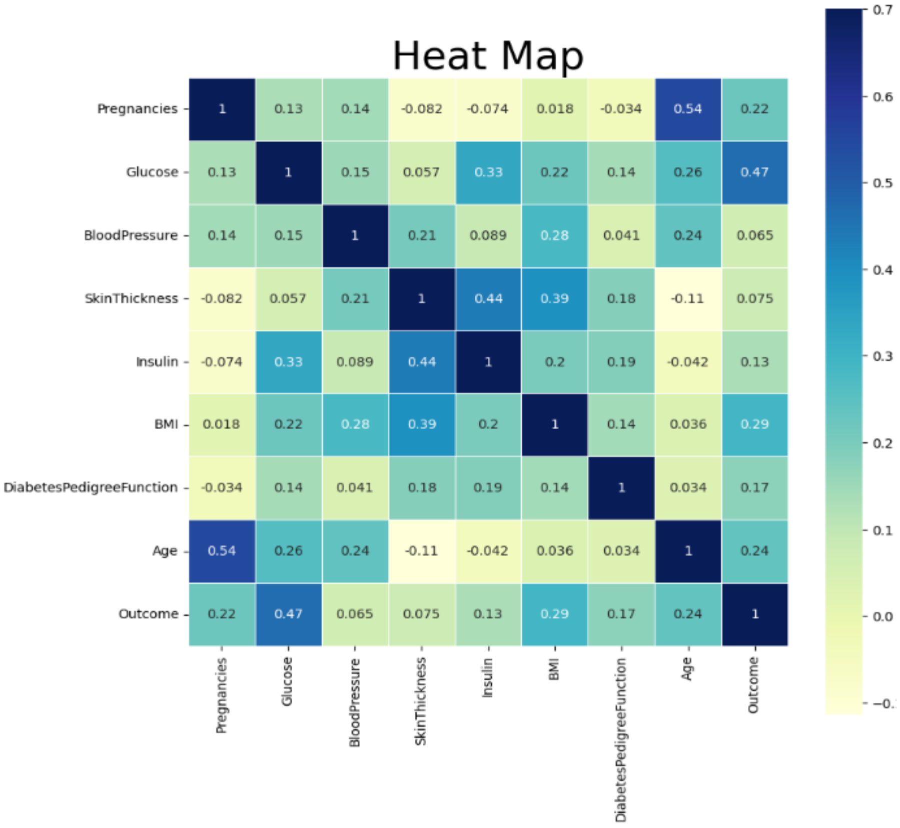


Figure 7-8: Gender and Diabetes Count plot Dataset 2

Correlation Matrix with Heatmap

A heat map is a two-dimensional representation of information with the help of colors. Heat maps can help the user visualize simple or complex information. The value of Pearson's Correlation Coefficient can be between -1 to +1. 1 means that they are highly correlated and 0 means no correlation.



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1	0.13	0.14	-0.082	-0.074	0.018	-0.034	0.54	0.22
Glucose	0.13	1	0.15	0.057	0.33	0.22	0.14	0.26	0.47
BloodPressure	0.14	0.15	1	0.21	0.089	0.28	0.041	0.24	0.065
SkinThickness	-0.082	0.057	0.21	1	0.44	0.39	0.18	-0.11	0.075
Insulin	-0.074	0.33	0.089	0.44	1	0.2	0.19	-0.042	0.13
BMI	0.018	0.22	0.28	0.39	0.2	1	0.14	0.036	0.29
DiabetesPedigreeFunction	-0.034	0.14	0.041	0.18	0.19	0.14	1	0.034	0.17
Age	0.54	0.26	0.24	-0.11	-0.042	0.036	0.034	1	0.24
Outcome	0.22	0.47	0.065	0.075	0.13	0.29	0.17	0.24	1

Figure 7-10: Heatmap Dataset 1

In Dataset 1, when observing the last row 'Outcome' and noticing its correlation scores with different features. It is noted that Glucose, BMI and Age are the most correlated with Outcome. BloodPressure, SkinThickness are the least correlated, hence they don't contribute much to the model so we can drop them. Glucose is the best indicator of diabetes outcome in this situation. It is seen that with strongly correlated features, the target class can be predicted more easily, and more meaningful results can be drawn.

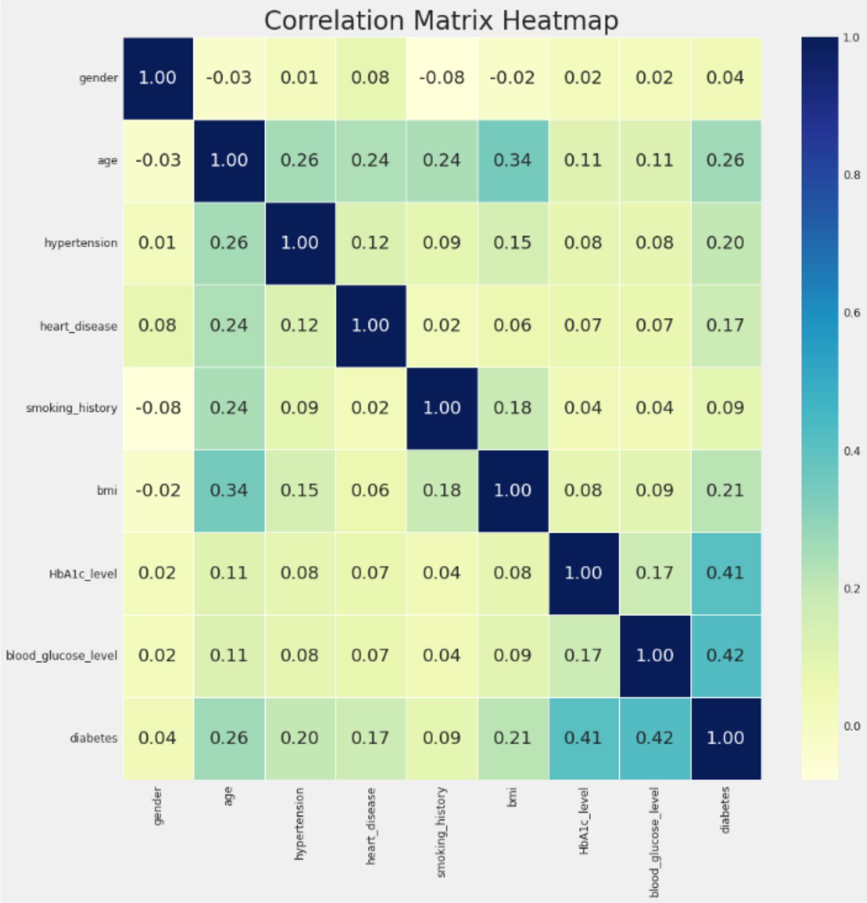


Figure 7-11: Heatmap Dataset 2

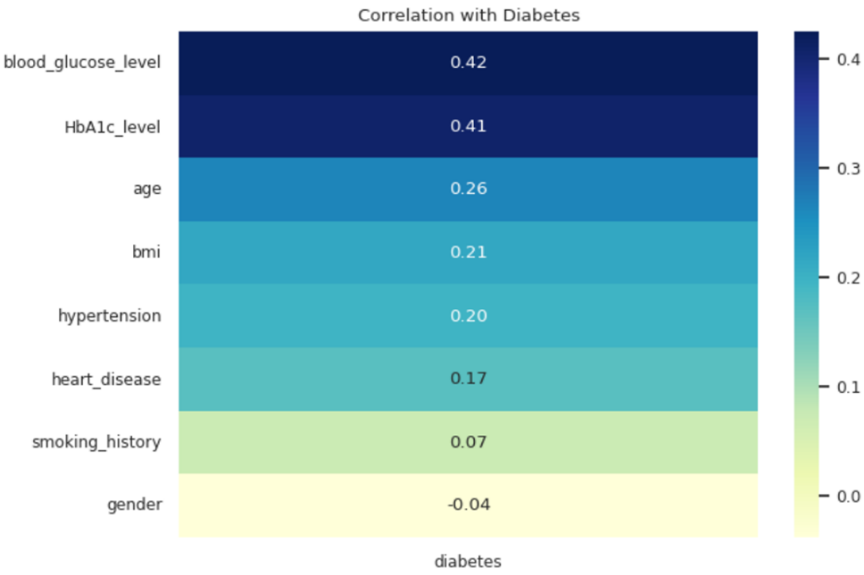


Figure 7-12: Diabetes Correlation

In Dataset 2, when observing the last row 'Diabetes' and noticing its correlation scores with different features. It is noted that HbA1c level, blood glucose level, hypertension, heart disease, bmi and age are the most correlated with Diabetes. smoking_history and gender have least importance. HbA1c level and Blood Glucose level are the best indicators of diabetes outcome in this situation.

7.2: Evaluation Metrics of ML Models

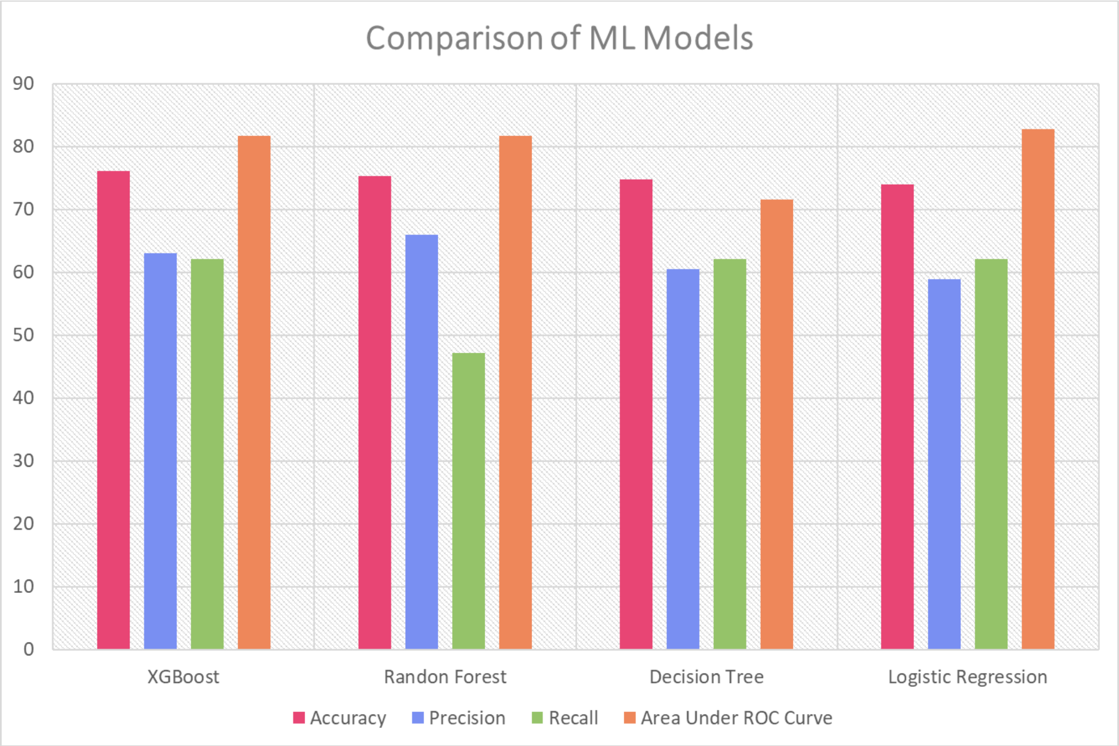


Figure 7-13: Comparison of ML Models Dataset 1

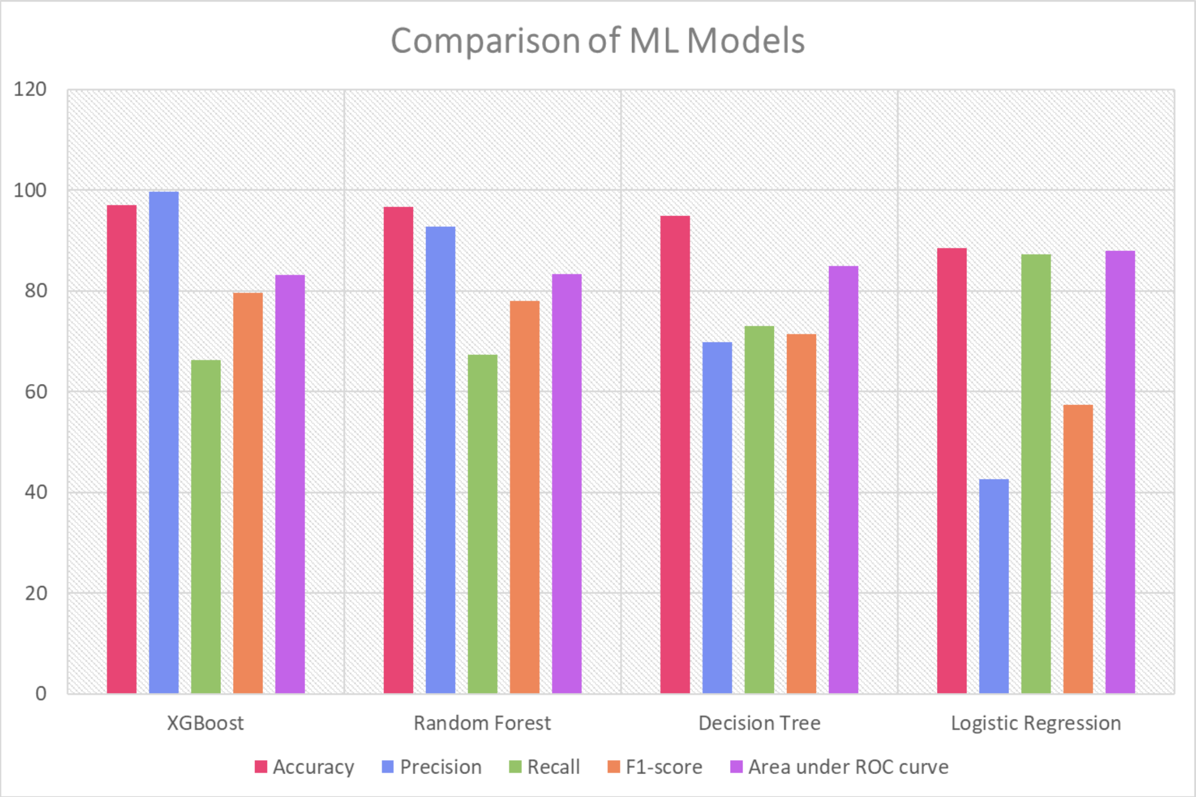
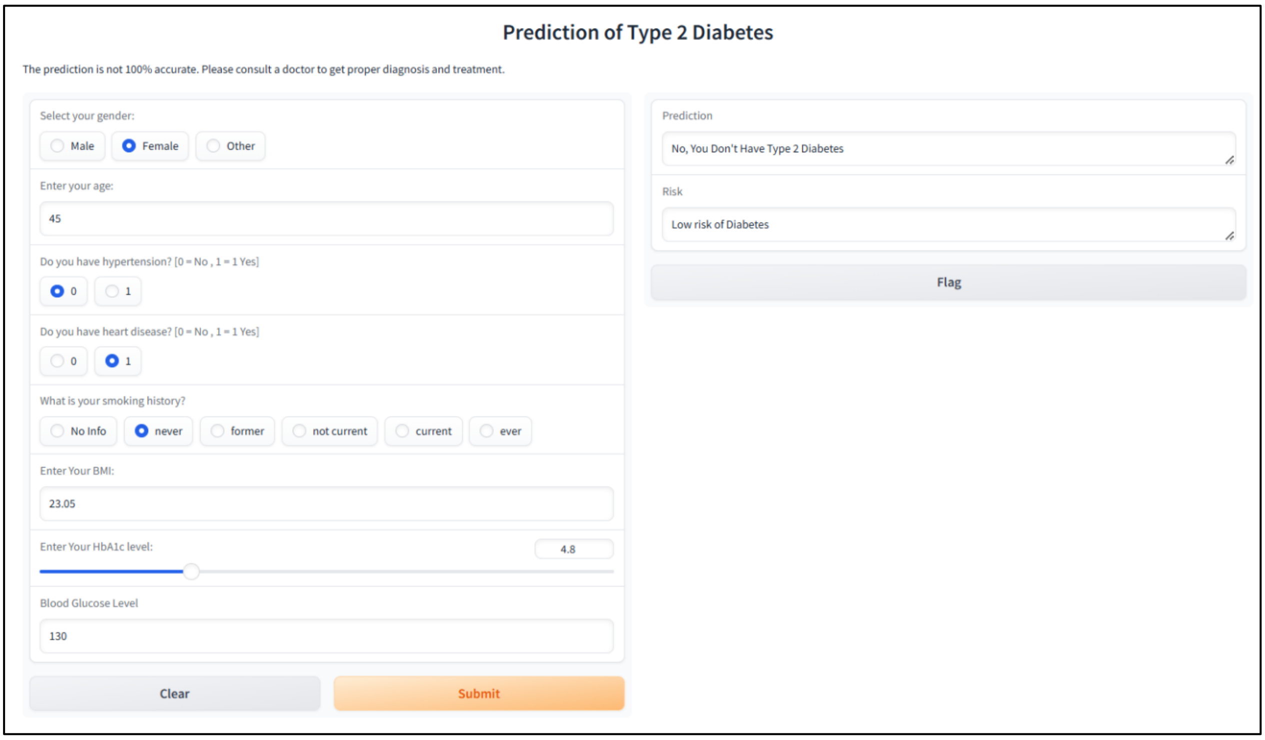


Figure 7-14: Comparison of ML Models Dataset 2

It can be inferred from the results that XGBoost model has the highest accuracy rate and performs well compared to other models. This result was observed in Dataset 1 and 2. As you can see, the highest accuracy rate attained by XGBoost model is 76% in Dataset 1 and is 97% in Dataset 2. Due to this reason and the presence of HbA1c level dataset column in Dataset 2, it is the most suitable dataset for Prediction of Type 2 diabetes.

7.3: GUI Output

I have created GUI using Gradio Interface in Google Colab. It resembles an online form. Now, users can enter values to determine if they have type 2 diabetes or not, based on certain medical data provided by the user. It can make almost accurate predictions and classify a person as having type 2 diabetes or not. It also predicts if a person has a risk of having prediabetes.



Prediction of Type 2 Diabetes

The prediction is not 100% accurate. Please consult a doctor to get proper diagnosis and treatment.

Select your gender:
○ Male ● Female ○ Other

Enter your age:
45

Do you have hypertension? [0 = No , 1 = 1 Yes]
● 0 ○ 1

Do you have heart disease? [0 = No , 1 = 1 Yes]
○ 0 ● 1

What is your smoking history?
○ No Info ● never ○ former ○ not current ○ current ○ ever

Enter Your BMI:
23.05

Enter Your HbA1c level:
4.8

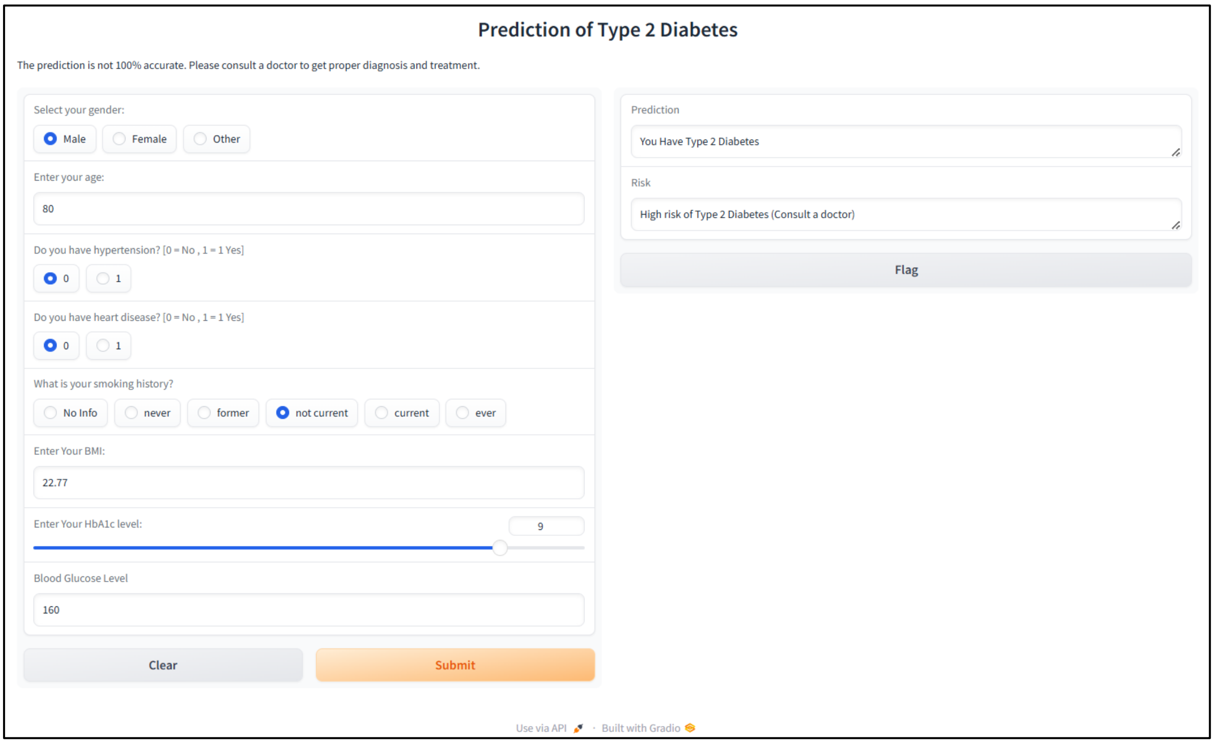
Blood Glucose Level
130

Prediction
No, You Don't Have Type 2 Diabetes

Risk
Low risk of Diabetes

Figure 7-15: Gradio GUI Output 1

Here, a user has entered his/her medical information that his/her gender is Female, she is 45 years old, don’t have hypertension but has heart disease, smoking history is never, BMI is 23.05, HbA1C level is 4.8 and blood glucose level is 130. The model accurately predicts that she does not have Type 2 Diabetes mostly based on BMI, HbA1C level and blood glucose level. It also predicts that she has low risk of diabetes.



Prediction of Type 2 Diabetes

The prediction is not 100% accurate. Please consult a doctor to get proper diagnosis and treatment.

Select your gender:
● Male ○ Female ○ Other

Enter your age:
80

Do you have hypertension? [0 = No , 1 = 1 Yes]
● 0 ○ 1

Do you have heart disease? [0 = No , 1 = 1 Yes]
● 0 ○ 1

What is your smoking history?
○ No Info ○ never ○ former ● not current ○ current ○ ever

Enter Your BMI:
22.77

Enter Your HbA1c level:
9

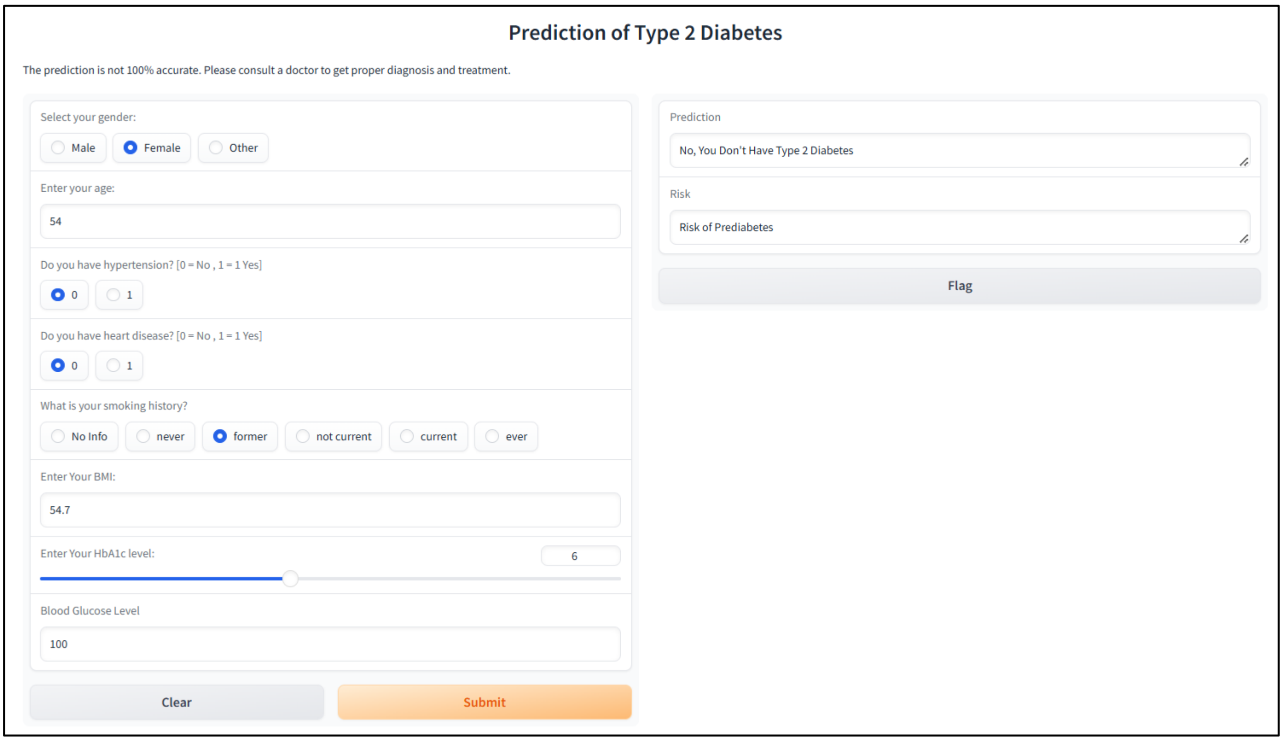
Blood Glucose Level
160

Prediction
You Have Type 2 Diabetes

Risk
High risk of Type 2 Diabetes (Consult a doctor)

Figure 7-16: Gradio GUI Output 2

Here, a user has entered his/her medical information that his/her gender is Male, he is 80 years old, don't have hypertension and heart disease, smoking history is not current, BMI is 22.77, HbA1C level is 9 and blood glucose level is 160. The model accurately predicts that he has Type 2 Diabetes mostly based on BMI, HbA1C level and blood glucose level. It predicts the risk also.



Prediction of Type 2 Diabetes

The prediction is not 100% accurate. Please consult a doctor to get proper diagnosis and treatment.

Select your gender:
○ Male ● Female ○ Other

Enter your age:
54

Do you have hypertension? [0 = No , 1 = 1 Yes]
● 0 ○ 1

Do you have heart disease? [0 = No , 1 = 1 Yes]
● 0 ○ 1

What is your smoking history?
○ No Info ○ never ● former ○ not current ○ current ○ ever

Enter Your BMI:
54.7

Enter Your HbA1c level:
6

Blood Glucose Level
100

Prediction
No, You Don't Have Type 2 Diabetes

Risk
Risk of Prediabetes

Figure 7-17: Gradio GUI Output 3

Here, a user has entered his/her medical information that his/her gender is Female, she is 54 years old, don’t have hypertension and heart disease, smoking history is former, BMI is 54.7, HbA1C level is 6 and blood glucose level is 100. The model accurately predicts that she does not have Type 2 Diabetes mostly based on BMI, HbA1C level and blood glucose level. It also predicts that she has a risk of prediabetes. This indicates that if she follows some preventive measures then she can reduce her chances of being affected by prediabetes or Type 2 Diabetes.

8 Discussion and Conclusion

People having 2 diabetes are very common in today's society. Symptoms are only noticeable gradually after the disease has progressed. HbA1c diagnostic test is used to identify prediabetes and type 2 diabetes. In this project, I have analyzed two Kaggle datasets which contains medical data regarding Type 2 diabetes. Pima Indian Dataset, that is, Dataset 1 contains medical data of only women, and the diagnostic test is Oral Glucose Tolerance Test. Diabetes Prediction Dataset contains medical data of men and women and the diagnostic test is HbA1c test . The most popular ML algorithms like Logistic Regression, Decision Tree, Random Forest, XGBoost have been used to build and train predictive models. In Dataset 1, The accuracy rate of ML models is : XGBoost – 76%, Random Forest – 75%, Decision Tree – 74%, Logistic Regression – 74%. In Dataset 2, The accuracy rate of ML models is : XGBoost – 96%, RF – 96%, DT – 94%, LR – 88%. So, XGB model has the highest accuracy rate for both Datasets. Due to the presence of HbA1c levels and XGB model accuracy of 97%, Dataset 2 was chosen for further development. Gradio Interface is used for GUI development. Users can enter medical information to determine if they have type 2 diabetes or not, and the ML model almost accurately predicts and classifies if the users have type 2 diabetes or not. It also predicts if a user has a risk of having prediabetes. This whole project has been implemented using Google Colaboratory because of ease of use, simplicity, and efficiency. After the project was completed, I found answers to the research questions in Chapter 3. The answer for the first question is that, building ML model to predict and classify whether a person has type 2 diabetes or not requires several procedures, like data preprocessing, data cleaning, exploratory data analysis, data visualization, feature selection, feature scaling, model selection, training, evaluation, and prediction. The dataset might contain null values, missing values, duplicate values, outliers, and class imbalance. All these issues should be resolved to build ML models with good performance. The answer to the second question is that it is possible to develop a classification model that predicts diabetes based on specific diagnostic metrics found in a dataset. The quality of the data, the selection of features, and the machine learning method all affect how well the model performs and how accurate it is. It must be confirmed that the dataset is clean (i.e., no missing or inconsistent values) and includes significant diagnostic metrics for accurate prediction. The answer to the third question is that, building a GUI (Graphical User Interface) for a machine learning model in Google Colaboratory can be achieved by using libraries such as Streamlit, Gradio, or Flask. I chose Gradio because it is a simple and efficient way to create UIs for machine learning models. Firstly, set up an environment in Google Colab by installing the necessary libraries, loading, and preprocessing the dataset and training the machine learning model. Secondly, create a GUI with Gradio Interface by installing and importing Gradio library, defining the prediction function and creating a Gradio interface with input and output components, to accept user input and display predictions. Even though, the implementation of the project has been successful, it faced many issues in the development. Sometimes, the model couldn’t make prediction if a user has no diabetes, prediabetes or type 2 diabetes, when the user entered HbA1c level value is 5.7 or 6.4. Gradio GUI would not appear sometimes because of Gradio library error issues. Handling medical data for machine learning purposes is challenging. The future works are further development of this project into a website or an app where a user can do a self-assessment and know when to consult a doctor if the disease worsens, be aware of all of types of diabetes, get dietary guidelines, treatment, ways to prevent the complications from occurring, tools to manage diabetes like medicine, water, meal reminder features and keeping a record of the blood sugar levels and HbA1c levels. It can be integrated into IoT devices like smartwatches. This machine learning project can be an asset to the healthcare industry and help in the implementation of AI and Machine Learning in Healthcare settings.

9 References

1) Mayo Clinic (2024) Diabetes - Symptoms and Causes. Available at: https://www.mayoclinic.org/diseases-conditions/diabetes/symptoms-causes/syc-20371444 (Accessed: 1 May 2024).
2) NVIDIA (no date) Machine Learning – What Is It and Why Does It Matter? - NVIDIA Data Science Glossary. Available at: https://www.nvidia.com/en-us/glossary/machine-learning/ (Accessed: 1 May 2024).
3) Tigga, N.P. and Garg, S., 2020. Prediction of type 2 diabetes using machine learning classification methods. Procedia Computer Science,167, pp.706-716.
4) Nadaf, S.S., Nikam, O.S., Zende, T.P., Pol, R.S. and Patil, P.D., Diabetes prediction in women using machine learning.
5) AKMEŞE, Ö.F., 2022. Diagnosing Diabetes with Machine Learning Techniques.
6) Hittite Journal of Science and Engineering, 9(1), pp.9-18.

7) Dey, S.K., Hossain, A. and Rahman, M.M., 2018, December. Implementation of a web application to predict diabetes disease: an approach using machine learning algorithm. In 2018 21st international conference of computer and information technology (ICCIT) (pp. 1-5). IEEE.

8) Cleveland Clinic (no date) Diabetes: What It Is, Causes, Symptoms, Treatment & Types, Cleveland Clinic. Available at: https://my.clevelandclinic.org/health/diseases/7104-diabetes (Accessed: May 16, 2024).

9) McLeod, S. (2023) Qualitative Vs Quantitative Research Methods & Data Analysis, Simply Psychology. Available at: https://www.simplypsychology.org/qualitative-quantitative.html

10) University of Southern California Libraries (2022) Research guides: Organizing your social sciences research paper: Quantitative methods, USC Libraries. Available at: https://libguides.usc.edu/writingguide/quantitative

11) Walch, K. (no date) Why Agile Methodologies Miss The Mark For AI & ML Projects, Forbes. Available at: https://www.forbes.com/sites/cognitiveworld/2020/01/19/why-agile-methodologies-miss-the-mark-for-ai--ml-projects/?sh=59ecd5c421ea (Accessed: 22 May 2024).

12) Sridharan, M. (2018) CRISP-DM - A Framework For Data Mining And Analysis, Think Insights. Available at: https://thinkinsights.net/data/crisp-dm/

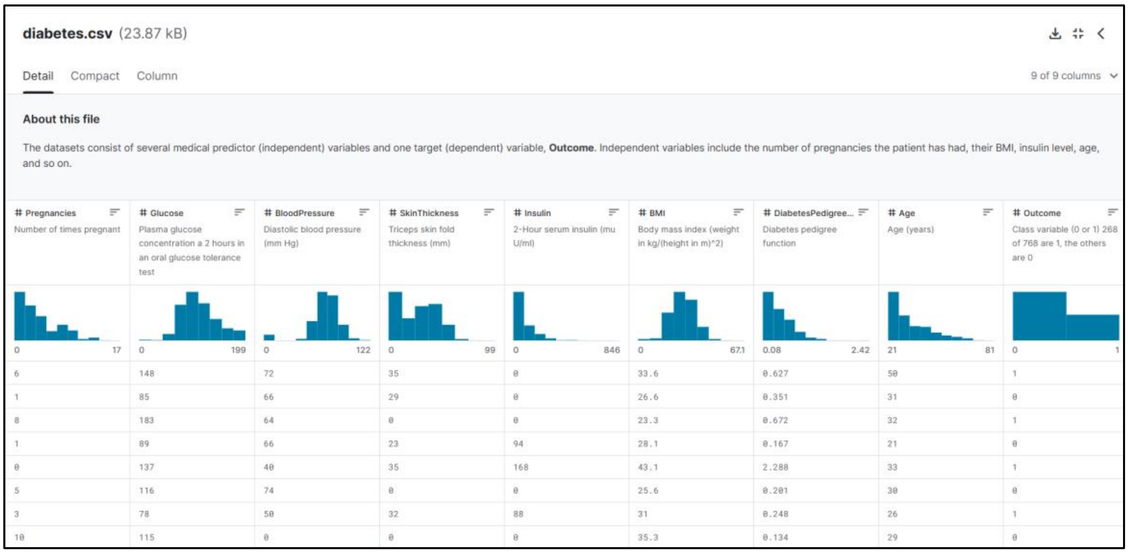
13) Life cycle of Machine Learning - Javatpoint (no date) www.javatpoint.com. Available at: https://www.javatpoint.com/machine-learning-life-cycle

14) Team, G. (no date) Quickstart, gradio.app. Available at: https://www.gradio.app/guides/quickstart

15) Watson, C. (2023) Type 2 Diabetes Mellitus: Symptoms, Diet, Medication, Treatment, Risk Factors, Definition, EZmed. Available at: https://www.ezmedlearning.com/blog/type-2-diabetes-mellitus-symptoms-medications

16) Keita, Z. (2022) Classification in Machine Learning: A Guide for Beginners, Datacamp. Available at: https://www.datacamp.com/blog/classification-machine-learning

10 Appendix



diabetes.csv (23.87 kB)

Detail Compact Column

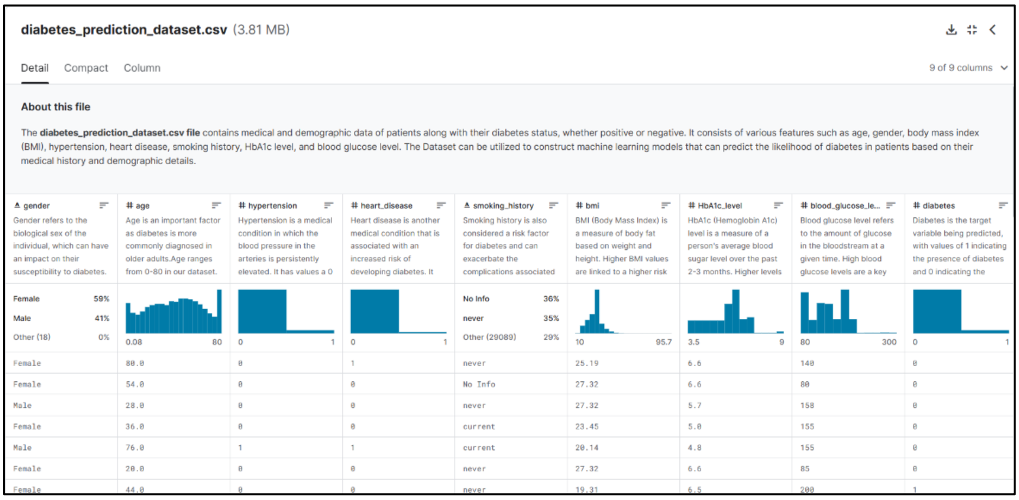
9 of 9 columns

About this file

The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, **Outcome**. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.

# Pregnancies	# Glucose	# BloodPressure	# SkinThickness	# Insulin	# BMI	# DiabetesPedigree...	# Age	# Outcome
Number of times pregnant	Plasma glucose concentration a 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 (weight in kg/(height in m)^2)	Diabetes pedigree function	Age (years)	Class variable (0 or 1) 268 of 768 are 1, the others are 0
0 17	0 199	0 122	0 99	0 846	0 67.1	0.08 2.42	21 81	0 1
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0

Figure 10-1: Dataset 1 Overview (Kaggle)



diabetes_prediction_dataset.csv (3.81 MB)

Detail Compact Column

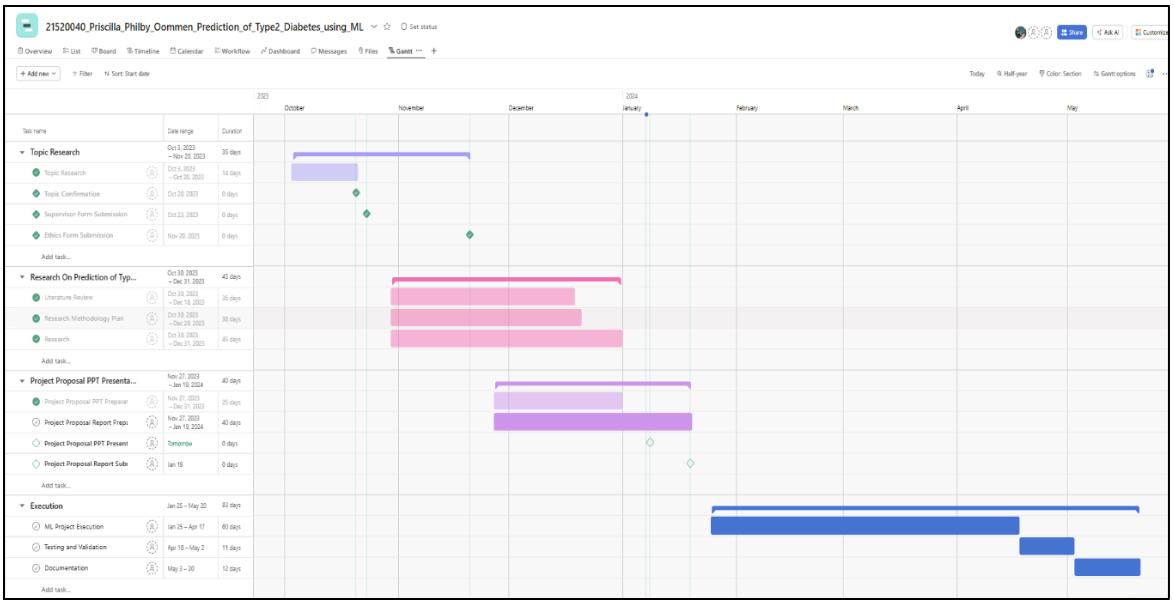
9 of 9 columns

About this file

The **diabetes_prediction_dataset.csv file** contains medical and demographic data of patients along with their diabetes status, whether positive or negative. It consists of various features such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level. The Dataset can be utilized to construct machine learning models that can predict the likelihood of diabetes in patients based on their medical history and demographic details.

gender	# age	# hypertension	# heart_disease	smoking_history	# bmi	# HbA1c_level	# blood_glucose_le...	# diabetes
Gender refers to the biological sex of the individual, which can have an impact on their susceptibility to diabetes.	Age is an important factor as diabetes is more commonly diagnosed in older adults.Age ranges from 0-80 in our dataset.	Hypertension is a medical condition in which the blood pressure in the arteries is persistently elevated. It has values a 0	Heart disease is another medical condition that is associated with an increased risk of developing diabetes. It	Smoking history is also considered a risk factor for diabetes and can exacerbate the complications associated	BMI (Body Mass Index) is a measure of body fat based on weight and height. Higher BMI values are linked to a higher risk	HbA1c (Hemoglobin A1c) level is a measure of a person's average blood sugar level over the past 2-3 months. Higher levels	Blood glucose level refers to the amount of glucose in the bloodstream at a given time. High blood glucose levels are a key	Diabetes is the target variable being predicted, with values of 1 indicating the presence of diabetes and 0 indicating the
Female 59% Male 41% Other (18) 0%	0.08 80	0 1	0 1	No Info 36% never 35% Other (29089) 29%	10 95.7	3.5 9	80 300	0 1
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1

Figure 10-2: Dataset 2 Overview (Kaggle)



21520040_Priscilla_Philby_Oommen_Prediction_of_Type2_Diabetes_using_ML

Task name	Date range	Duration
Topic Research	Oct 3, 2023 – Nov 20, 2023	35 days
Topic Research	Oct 3, 2023 – Oct 20, 2023	14 days
Topic Confirmation	Oct 20, 2023	0 days
Supervisor Form Submission	Oct 23, 2023	0 days
Ethics Form Submission	Nov 20, 2023	0 days
Add task...		
Research On Prediction of Typ...	Oct 30, 2023 – Dec 31, 2023	45 days
Literature Review	Oct 30, 2023 – Dec 18, 2023	36 days
Research Methodology Plan	Oct 30, 2023 – Dec 20, 2023	38 days
Research	Oct 30, 2023 – Dec 31, 2023	45 days
Add task...		
Project Proposal PPT Presenta...	Nov 27, 2023 – Jan 19, 2024	40 days
Project Proposal PPT Preparat	Nov 27, 2023 – Dec 31, 2023	25 days
Project Proposal Report Prep:	Nov 27, 2023 – Jan 19, 2024	40 days
Project Proposal PPT Present	Tomorrow	0 days
Project Proposal Report Subı	Jan 19	0 days
Add task...		
Execution	Jan 25 – May 20	83 days
ML Project Execution	Jan 25 – Apr 17	60 days
Testing and Validation	Apr 18 – May 2	11 days
Documentation	May 3 – 20	12 days
Add task...		

Figure 10-3: Project Gantt Chart