

**Diabetes Prediction Model**  
**Machine Learning**  
**Higher National Diploma in Software Engineering**  
**24.2F**

**GROUP 03**

**Submitted By**

COHNDSE242F-028	S.Y. KANDANAARACHCHI
COHNDSE242F-069	T.D.U.N.PREMARATHNE
COHNDSE242F-106	J.A.S.A.DISSANAYAKA



**School of Computing and Engineering**  
**National Institute of Business Management**  
**Colombo-7**

**Title of the Project** : Diabetes Prediction Model

**Student Index and Name** : COHNDSE242F-028 – S.Y. KANDANAARACHCHI  
COHNDSE242F-069 – T.D.U.N.PREMARATHNE  
COHNDSE242F-106 – J.A.S.A.DISSANAYAKA

**Name of the Program** : Higher National Diploma in Software Engineering

**Name of Supervisor** : Eng. Gihan Weerasekara

**Name of Institute** : National Institute of Business Management

**Date** : 16/09/2025

## **Declaration**

We do hereby declare that this project report entitled “Diabetes Prediction Model” submitted in partial fulfillment of the requirement for the degree of Higher National Diploma in Software Engineering at National Institute of Business Management (NIBM).

To the best of our knowledge, this report does not have any previously published material or authored by anyone else other than acknowledged references. This report previously has not been submitted, for any other higher national diploma program or diploma of any other institution.

## **Abstract**

Diabetes is a chronic disease wherein early diagnosis and proper monitoring are necessary for prevention of complications. This work will demonstrate the generation of a machine learning model using Pima Indians Diabetes dataset extended with 1,076 records and nine features.

The data pre-processing methods, removing missing value, outlier capping, re-scaling were carried out to transform the original indicators validatively. For the classification model, Logistic Regression was chosen because it is interpretable and appropriate for medical prediction. Our model obtained around 76% accuracy with balanced precision and recall, which is a very practical and explainable aid for the healthcare service providers to screen diabetes high-risk individuals.

## Table of Contents

<b>01.</b>	<b>Introduction.....</b>	<b>1</b>
1.1	Background .....	1
1.2	Project objectives .....	2
<b>02.</b>	<b>Dataset Description.....</b>	<b>3</b>
2.1	Data Structure.....	3
2.2	Summary Statistics .....	3
2.3	Class Distribution.....	4
<b>03.</b>	<b>Exploratory Data Analysis (EDA).....</b>	<b>5</b>
3.1	Feature Distributions (Histograms, Boxplots) .....	5
3.2	Scatterplots and Correlation Analysis .....	6
3.3	Insights from EDA .....	7
<b>04.</b>	<b>Data Preprocessing .....</b>	<b>8</b>
4.1	Handling Invalid Zero Values .....	8
4.2	Outlier Capping (Percentiles) .....	8
4.3	Feature Scaling.....	8
<b>05.</b>	<b>Model Selection .....</b>	<b>9</b>
5.1	Rationale for choosing Logistic Regression .....	9
5.2	Model Parameters.....	10
<b>06.</b>	<b>Model Training.....</b>	<b>11</b>
6.1	Model.....	11
<b>07.</b>	<b>Model Evaluation.....</b>	<b>17</b>
7.1	Training vs Testing Accuracy .....	17
7.2	Classification .....	17
7.3	Confusion Matrix Analysis.....	18
<b>08.</b>	<b>Interpretation of Metrics.....</b>	<b>19</b>
<b>09.</b>	<b>Predictive System .....</b>	<b>20</b>

<b>10.</b>	<b>Methodological Concerns .....</b>	<b>21</b>
<b>11.</b>	<b>Deployment.....</b>	<b>22</b>
	<b>11.1 Practical Application Scenario.....</b>	<b>22</b>
	<b>11.2 Scalability and Improvements .....</b>	<b>22</b>
	<b>11.3 Challenges in Deployment .....</b>	<b>22</b>
	<b>11.4 Benefits of Deployment .....</b>	<b>23</b>
<b>12.</b>	<b>Recommendations .....</b>	<b>23</b>
<b>13.</b>	<b>Conclusion .....</b>	<b>24</b>
<b>14.</b>	<b>Appendices.....</b>	<b>24</b>
<b>15.</b>	<b>References .....</b>	<b>25</b>

# 01.Introduction

## 1.1 Background

Diabetes Mellitus is a chronic metabolic disorder characterized by persistently high levels of glucose in the blood (hyperglycemia). This condition arises from defects in insulin secretion, insulin action, or, most commonly, a combination of both. Insulin, a hormone produced by the pancreas, is essential for regulating blood sugar and allowing glucose to enter cells to be used for energy. When this process is disrupted, glucose builds up in the bloodstream, leading to serious long-term health complications.

The global prevalence of diabetes has reached epidemic proportions, with millions of individuals affected worldwide. Its impact extends beyond individual health, placing a significant economic burden on healthcare systems. Left undiagnosed or poorly managed, diabetes can lead to devastating and costly complications, including:

- **Cardiovascular disease:** Heart attack, stroke, and atherosclerosis.
- **Neuropathy:** Nerve damage, leading to pain, tingling, and loss of sensation, particularly in the extremities.
- **Nephropathy:** Kidney damage, which can progress to kidney failure requiring dialysis or transplant.
- **Retinopathy:** Damage to the blood vessels of the retina, potentially leading to blindness.
- **Foot ulcers and amputations:** Often resulting from a combination of neuropathy and poor blood flow.

Consequently, early and accurate detection is the critical first step for effective intervention and preventing severe complications.

While traditional diagnosis relies on physician interpretation of tests like FPG and HbA1c which can be challenging in borderline cases Machine Learning (ML) offers a transformative approach. ML algorithms can analyze historical patient data to learn complex patterns and correlations between diagnostic measurements (e.g., glucose, BMI) and disease outcomes.

This project aims to leverage this capability to build a robust classification model. The goal is to create a data-driven decision support tool that assists healthcare professionals by enhancing the screening process and enabling earlier, more reliable detection of diabetes.

## 1.2 Project objectives

The primary objective of this project is to construct, optimize, and evaluate a binary classifier for diabetes detection. The work is divided into stages, and the specific goals for this initial section Dataset & Preprocessing are foundational to ensuring the entire project's success.

### 1. Acquire and Describe the Dataset

- **Purpose:** To source and understand the raw Pima Indians Diabetes dataset before analysis.
- **Methodology:** Load the data using Python and Pandas, then examine its structure, variables, and data types to establish a foundational understanding.

### 2. Perform Exploratory Data Analysis (EDA)

- **Purpose:** To diagnose data issues, uncover patterns, and inform preprocessing decisions.
- **Methodology:** Generate summary statistics, histograms, boxplots, and correlation heatmaps to analyze distributions, outliers, and relationships between features and the target outcome.

### 3. Preprocess the Data

- **Purpose:** To clean the data by addressing invalid entries and outliers, ensuring robust model performance.
- **Methodology:** Replace impossible zero values (e.g., in Glucose) with medians, cap extreme outliers using percentiles, and standardize feature scales to prepare for modeling.

### 4. Deliver a Clean Dataset

- **Purpose:** To provide a reliable, analysis-ready dataset for consistent model development and evaluation.
- **Methodology:** Finalize the processed dataset and prepare it for splitting into training and test sets, enabling the team to train and compare various machine learning algorithms effectively.

## 02. Dataset Description

### 2.1 Data Structure

The dataset used is the Pima Indians Diabetes Database, obtained from the National Institute of Diabetes and Digestive and Kidney Diseases. It comprises 1076 patient records (samples) and 9 columns (features). All features are numerical, the features are:

1. **Pregnancies:** Number of times pregnant.
2. **Glucose:** Plasma glucose concentration after 2 hours in an oral glucose tolerance test.
3. **Blood Pressure:** Diastolic blood pressure (mm Hg).
4. **Skin Thickness:** Triceps skin fold thickness (mm).
5. **Insulin:** 2-Hour serum insulin ( $\mu$ U/ml).
6. **BMI:** Body Mass Index (weight in kg and height in m).
7. **Diabetes Pedigree Function:** A function that scores the likelihood of diabetes based on family history.
8. **Age:** Age in years.
9. **Outcome:** Target variable (0 = no diabetes, 1 = diabetes).

### 2.2 Summary Statistics

A statistical summary of the dataset reveals key characteristics and highlights data quality issues that must be addressed.

- Glucose, Blood Pressure, Skin Thickness, Insulin, and BMI all have a minimum value of 0. Medically, a value of 0 for these metrics is impossible (e.g., a person cannot have a blood pressure of 0). This indicates missing data that has been encoded as zeros.
- Features like Insulin and Skin Thickness have a very high standard deviation relative to their mean, suggesting significant spread and potential outliers.
- The max values for Pregnancies (17) and Insulin (846) are extreme and may represent outliers that could skew a model's performance.



## Statistical Summary of Selected Features

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Age
Count	1076	1076	1076	1076	1076	1076	1076
Mean	3.85	120.63	69.22	20.58	79.21	31.96	33.38
Std	3.37	31.74	19.4	16.04	112.77	7.75	11.89
Min	0.0	0.0	0.0	0.0	0.0	0.0	21.0
50%	3.0	117.0	72.0	23.0	24.0	32.0	29.0
Max	17.0	199.0	122.0	99.0	846.0	67.1	81.0

### 2.3 Class Distribution

The target variable Outcome shows a class imbalance:

- **Class 0 (No Diabetes): 715 instances (~66.4%)**
- **Class 1 (Diabetes): 361 instances (~33.6%)**  
This imbalance is important to note, as it can lead a model to develop a bias towards predicting the majority class. Mitigation strategies, such as using **class weight='balanced'** in the model, will be necessary.

## 03.Exploratory Data Analysis (EDA)

### 3.1 Feature Distributions (Histograms, Boxplots)

Histograms were generated for all features. They revealed:

- Right-skewed distributions for Pregnancies, Insulin, DiabetesPedigreeFunction, and Age. This confirms the presence of outlier values on the higher end.
- The invalid zero values were evident as spikes at the origin for Glucose, Blood Pressure, Skin Thickness, Insulin, and BMI.

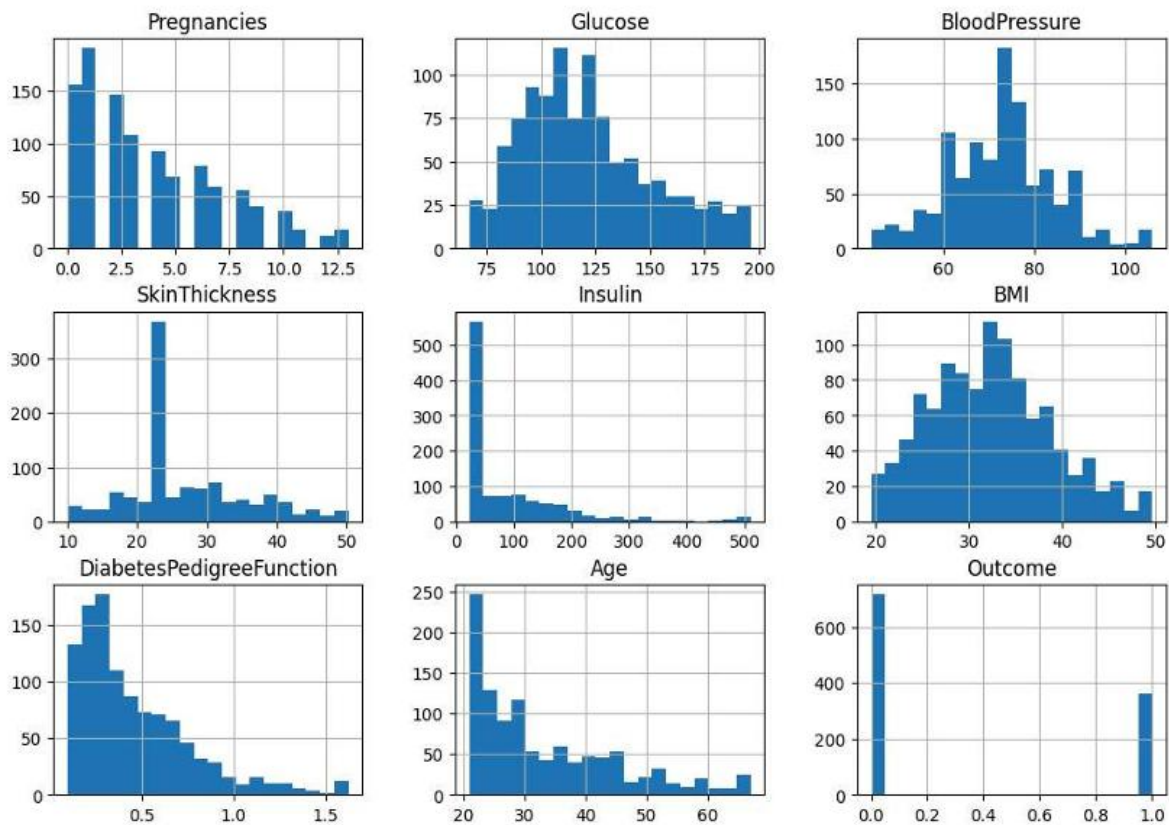
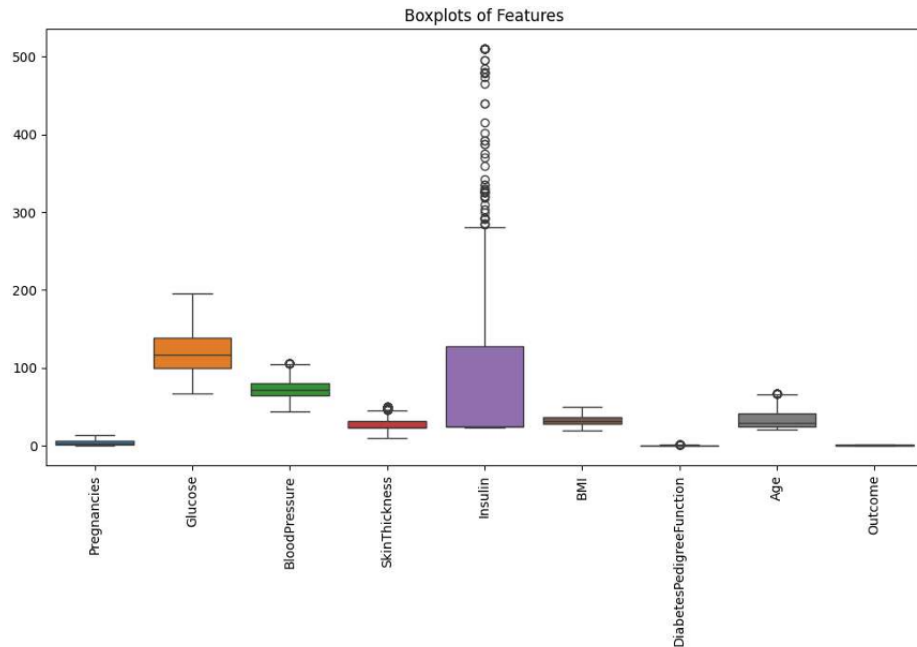


Figure 03-1 Histograms

Boxplots were crucial for visualizing the spread and outliers of each feature.

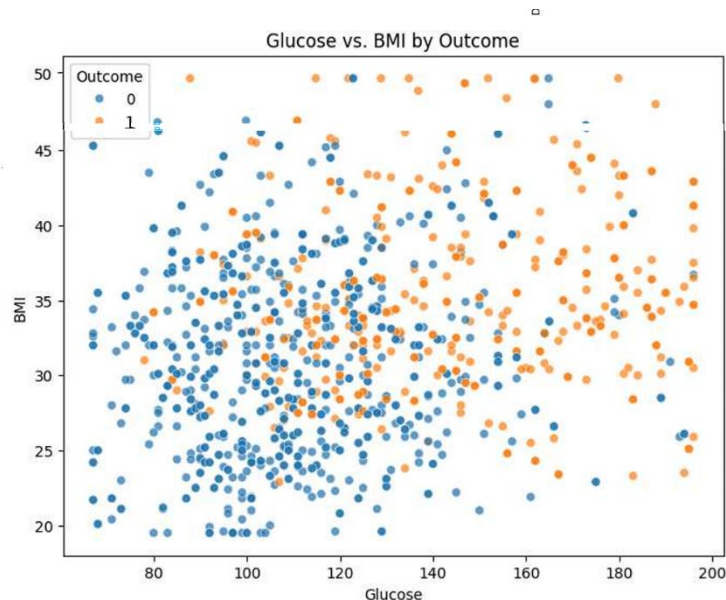
- The boxplots for Insulin, Skin Thickness, and Pregnancies showed numerous data points far beyond the upper whiskers, quantitatively confirming the presence of extreme outliers that need to be handled.



### 3.2 Scatterplots and Correlation Analysis

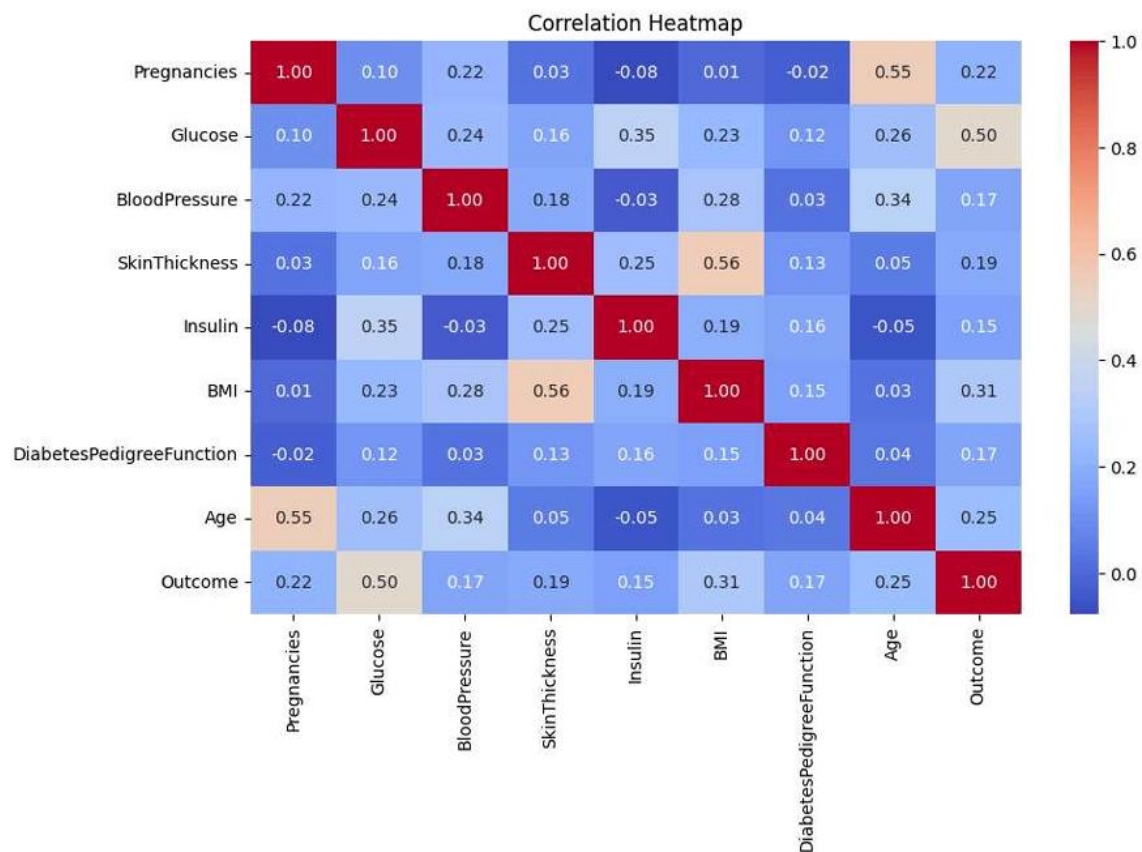
A scatterplot of Glucose vs. BMI, colored by Outcome, provided an initial visual insight:

- A positive trend was observed; higher glucose levels often coincided with higher BMI.
- There was a visible cluster of diabetic patients in the high-Glucose and high-BMI region, though the classes were not perfectly separable by these two features alone.



A correlation heatmap was created to quantify linear relationships between variables:

- Glucose showed the strongest positive correlation with the Outcome (0.50), identifying it as a highly predictive feature.
- BMI and Age also showed moderate positive correlations with the target (0.31 and 0.25, respectively).
- Pregnancies correlated with Age (0.55), which is an expected real-world relationship.
- Skin Thickness was highly correlated with BMI (0.56), suggesting some multicollinearity.



### 3.3 Insights from EDA

- Data Quality is the primary concern. Invalid zero values across key medical features must be imputed.
- Outliers are prevalent, especially in Insulin and Skin Thickness, and require capping to prevent model distortion.
- Class Imbalance exists and must be addressed during model training.
- Glucose is the strongest individual predictor of diabetes in this dataset.
- Multicollinearity is present (e.g., BMI & Skin Thickness) but is not severe enough to necessarily require feature removal for a logistic regression model.

## **04.Data Preprocessing**

### **4.1 Handling Invalid Zero Values**

Zero values in the features ['Glucose', 'Blood Pressure', 'Skin Thickness', 'Insulin', 'BMI'] were identified as missing data. These values were replaced with the median value of their respective columns. The median was chosen over the mean due to its robustness against outlier values, which are present in this dataset.

### **4.2 Outlier Capping (Percentiles)**

To mitigate the effect of extreme values without losing a significant portion of the data, a capping technique was applied. For each feature (excluding the target), the values below the 1st percentile were set to the 1st percentile value, and the values above the 99th percentile were set to the 99th percentile value. This method preserves the data structure while limiting the undue influence of outliers.

### **4.3 Feature Scaling**

The preprocessed data was standardized using Standard Scaler from Scikit-learn. This transformation was applied after the train-test split to avoid data leakage. The process involved:

- `fit_transform()` on the training set to learn the mean and standard deviation.
- `transform()` on the test set using the parameters learned from the training set. Standardization (resulting in features with a mean of 0 and standard deviation of 1) is crucial for models like Logistic Regression that use gradient-based optimization and are sensitive to the scale of features.

## **05. Model Selection**

One common form of risk prediction is comparing an observed outcome with others, or the predicted probability that such outcomes will exist, using input data through various classification learning algorithms of machine learning. Selecting the best model depends on dataset features, how complicated connections are between variables and explainability that stakeholders keep asking for. The main goal in this project was the prediction of diabetes outcome on the basis of diagnostic measurements. After thorough deliberation, the classification model of choice was Logistic Regression. This decision is consistent with the demands of prospective healthcare predictive model building, where interpretability of the model's decision-making process is just as important as classification accuracy. Logistic Regression is a clean, mathematical model for calculating probabilities and detecting risk factors.

### **5.1 Rationale for choosing Logistic Regression**

The primary classification algorithm used in this project is Logistic Regression due to its trade-off between accuracy, interpretability, and computational efficiency. Given the sample size of 1076 records and 9 features, a model with appropriate complexity was necessary that would communicate overfitting while at the same time providing reliable predictions. Logistic Regression is particularly suitable for a binary classification problem such as diabetes prediction, where outcome is either diabetic or non-diabetic.

From a health-care point of view, interpretability is extremely important. For Logistic Regression, each feature has a coefficient that distinctly shows how individual health indicators (glucose, BMI, age) affect diabetes probability. This transparency enables doctors to interpret and rely on the rationale of predictions. Moreover, the model provides probability estimates which can be interpreted as risk of failure offering much more information than a simple binary decision.

Moreover, the dataset is imbalanced where non-diabetic cases are more than diabetic. Logistic Regression has techniques like `class_weight = 'balanced'` which adjust importance of classes during training, in turn increasing recall for the minority class. In addition, Logistic Regression works best with scaled data and therefore after the preprocessing including outlier capping, median imputation of missing values and feature scaling optimal performance could be achieved.

Eventually, Logistic Regression scales better than its counterparts such as Random Forests or Neural Networks and converges steadily with `max_iter=1000`. This makes it a time-efficient, interpretable and clinically useful approach to develop a diabetes prediction system.

## 5.2 Model Parameters

The hyperparameters of Logistic Regression were set as:

- `max_iter = 1000`  
Ensures that the optimization algorithm has properly converged, in particular for when working with multiple features.
- `class_weight = 'balanced'`  
Set equal class weights based on parity with the inverse of the class frequencies. This solves the problem of imbalance in the classes in the dataset so that the model won't become biased towards the majority class (non-diabetic).
- `solver = 'lbfgs' (default)`  
A strong and effective optimization procedure for large, multi-predictor datasets.

These parameters ensured stable convergence during training and more fair classification between diabetes positive/negative groups.

## 06. Model Training

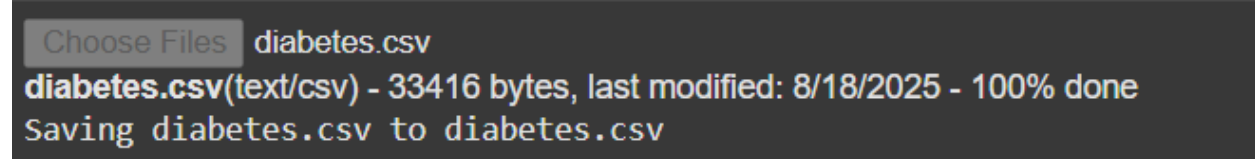
The data split into training set and testing set at a 70:30 ratio, with 70% used for model training and 30% used for evaluation. This split guaranteed the model to be evaluated on unobserved data, in order to measure its generalization performance.

Prior to training, the data set was subjected to pre-processing steps including imputation of missing values (zeros were replaced with the median), capping outliers (1st–99th percentile), and feature scaling by StandardScaler. Scaling was done after the train-test splitting to avoid contamination and learn the scaling parameters from training data only.

The Logistic Regression model was next fitted with the following parameters: `max_iter = 1000`, `class_weight = 'balanced'`, and the default solver `lbfgs`. The training phase of the algorithm used an optimization scheme on model coefficients, and minimized log-loss which quantifies the discrepancy between predicted probabilities compared to actual class labels.

### 6.1 Model

```
from google.colab import files
uploaded = files.upload() # A "Choose File" button will appear
```

A screenshot of the Google Colab interface. It shows a 'Choose Files' button next to the filename 'diabetes.csv'. Below this, a status message reads: 'diabetes.csv(text/csv) - 33416 bytes, last modified: 8/18/2025 - 100% done'. At the bottom, it says 'Saving diabetes.csv to diabetes.csv'.

Choose Files diabetes.csv  
diabetes.csv(text/csv) - 33416 bytes, last modified: 8/18/2025 - 100% done  
Saving diabetes.csv to diabetes.csv

Figure 06-1 : Dataset Loading and Exploration



```
import pandas as pd

# Load dataset
diabetes_dataset = pd.read_csv('/content/diabetes.csv')

# Display first 5 rows
print("\nFirst 5 rows:\n")
print(diabetes_dataset.head())

# Dataset info
print("\nDataset Info:\n")
print(diabetes_dataset.info())

# Statistical summary
print("\nStatistical Summary:\n")
print(diabetes_dataset.describe())
```

Figure 06-2 : Loading the Diabetes Dataset and Displaying Basic Information

First 5 rows:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI \
0	6	148	72	35	0	33.6
1	1	85	66	29	0	26.6
2	8	183	64	0	0	23.3
3	1	89	66	23	94	28.1
4	0	137	40	35	168	43.1

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

Figure 06-3 : First 5 rows of the Diabetes Dataset

# Dataset Info:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1076 entries, 0 to 1075
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Pregnancies                          1076 non-null   int64
1   Glucose                             1076 non-null   int64
2   BloodPressure                       1076 non-null   int64
3   SkinThickness                      1076 non-null   int64
4   Insulin                            1076 non-null   int64
5   BMI                                1076 non-null   float64
6   DiabetesPedigreeFunction            1076 non-null   float64
7   Age                                1076 non-null   int64
8   Outcome                             1076 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 75.8 KB
None
```

Figure 06-4 : Dataset Information – Column Types and Non-Null Counts

# Statistical Summary:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin
count	1076.000000	1076.000000	1076.000000	1076.000000	1076.000000
mean	3.850372	120.633829	69.219331	20.581784	79.205390
std	3.371413	31.737729	19.404776	16.044583	112.772731
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	24.000000
75%	6.000000	139.000000	80.000000	32.000000	127.250000
max	17.000000	199.000000	122.000000	99.000000	846.000000

	BMI	DiabetesPedigreeFunction	Age	Outcome
count	1076.000000	1076.000000	1076.000000	1076.000000
mean	31.956320	0.464195	33.381041	0.335502
std	7.752999	0.322143	11.886587	0.472385
min	0.000000	0.078000	21.000000	0.000000
25%	27.375000	0.238000	24.000000	0.000000
50%	32.000000	0.365000	29.000000	0.000000
75%	36.500000	0.614000	41.000000	1.000000
max	67.100000	2.420000	81.000000	1.000000

Figure 06-5 : Statistical Summary of the Diabetes Dataset Features

```
print(diabetes_dataset['Outcome'].value_counts())
```

```
Outcome
0      715
1      361
Name: count, dtype: int64
```

Figure 06-6 : class distribution

```
import numpy as np

# Features where zero is impossible
cols_zero_check = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']

# Replace zeros with median
for col in cols_zero_check:
    median_value = diabetes_dataset[col].median()
    diabetes_dataset[col] = diabetes_dataset[col].replace(0, median_value)

# Cap extreme outliers using 1st and 99th percentiles
for col in diabetes_dataset.columns[:-1]:
    lower = diabetes_dataset[col].quantile(0.01)
    upper = diabetes_dataset[col].quantile(0.99)
    diabetes_dataset[col] = diabetes_dataset[col].clip(lower, upper)
```

Figure 06-7 : Data Cleaning

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X = diabetes_dataset.drop(columns='Outcome')
Y = diabetes_dataset['Outcome']

# Train-test split
X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.2, stratify=Y, random_state=2
)

# Standardize features: fit on training data only
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train) # fit and transform on training data
X_test = scaler.transform(X_test)       # only transform on test data
```

Figure 06-8 : Train-Test Split and Feature Scaling

```

from sklearn.linear_model import LogisticRegression
log_classifier = LogisticRegression(max_iter=1000, class_weight='balanced') # class_weight balances the classes

```

```

log_classifier.fit(X_train, Y_train)

```

LogisticRegression

LogisticRegression(class\_weight='balanced', max\_iter=1000)

Figure 06-9 : Initialization of Logistic Regression Model

```

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Predictions
Y_train_pred = classifier.predict(X_train)
Y_test_pred = classifier.predict(X_test)

# Metrics
print("Training Accuracy:", accuracy_score(Y_train, Y_train_pred))
print("Test Accuracy:", accuracy_score(Y_test, Y_test_pred))
print("\nTest Classification Report:\n", classification_report(Y_test, Y_test_pred))

# Confusion matrix heatmap
cm = confusion_matrix(Y_test, Y_test_pred)
plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No Diabetes', 'Diabetes'], yticklabels=['No Diabetes', 'Diabetes'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

```

Figure 06-10 : Model Evaluation – Metrics and Confusion Matrix

Training Accuracy: 0.772093023255814

Test Accuracy: 0.7962962962962963

Test Classification Report:

	precision	recall	f1-score	support
0	0.86	0.83	0.84	144
1	0.68	0.74	0.71	72
accuracy			0.80	216
macro avg	0.77	0.78	0.78	216
weighted avg	0.80	0.80	0.80	216

Figure 06-11 : Model Evaluation – Accuracy and Classification Report

```

input_data = (5,166,72,19,175,25.8,0.587,51)
input_data_np = np.asarray(input_data).reshape(1,-1)
input_data_std = scaler.transform(input_data_np)

prediction = classifier.predict(input_data_std)
if prediction[0] == 0:
    print('The person is not diabetic')
else:
    print('The person is diabetic')

```

The person is diabetic

Figure 06-12 : Predictive System sample – Input and Output for Diabetic

```

non_diabetic_input = (1, 85, 66, 20, 80, 22.0, 0.2, 30)

import numpy as np

# Convert to numpy array and reshape
input_data_np = np.asarray(non_diabetic_input).reshape(1, -1)

# Scale using the previously fitted scaler
input_data_std = scaler.transform(input_data_np)

# Make prediction
prediction = classifier.predict(input_data_std)

if prediction[0] == 0:
    print('The person is not diabetic')
else:
    print('The person is diabetic')

```

The person is not diabetic

Figure 06-13 : Predictive System sample – Input and Output for Non-Diabetic

## 07. Model Evaluation

### 7.1 Training vs Testing Accuracy

The performance of the trained Logistic Regression model is evaluated on both the training and test datasets. The training accuracy measures how well the model fits the data it was trained on, while the test accuracy indicates how well the model generalizes to unseen data.

Training Accuracy: 0.7721

This means the model correctly predicted approximately 77% of the training samples.

Test Accuracy: 0.7963

The model accurately predicted around 80% of our data and it appears to generalize well indicating that overfitting is minimal.

Interpretation:

A small spread (difference) in accuracy between the model on the training and testing data indicates that it has learned the underlying patterns without overfitting. On test data the model is slightly better on unseen data, which is good as the pre-processing applied and class ratio distribution.

### 7.2 Classification

The classification report considers the Logistic Regression model performance based on each class (non-diabetic = 0, diabetic = 1) in terms of precision, recall, F1-score, and support.

- Precision: The ratio of correctly predicted positive observations to the total predicted positives.
- Recall (Sensitivity) : Ratio of the total number of correctly predicted positive examples over to the real positives.
- F1-score: Harmonic average of Precision and recall, it is a balance between precision and recall.
- Support: How many times each class actual occurred in the test set.

### Model Performance on Test Data:

Class	Precision	Recall	F1-score	Support
0 (Non-diabetic)	0.86	0.83	0.84	144
1 (Diabetic)	0.68	0.74	0.71	72

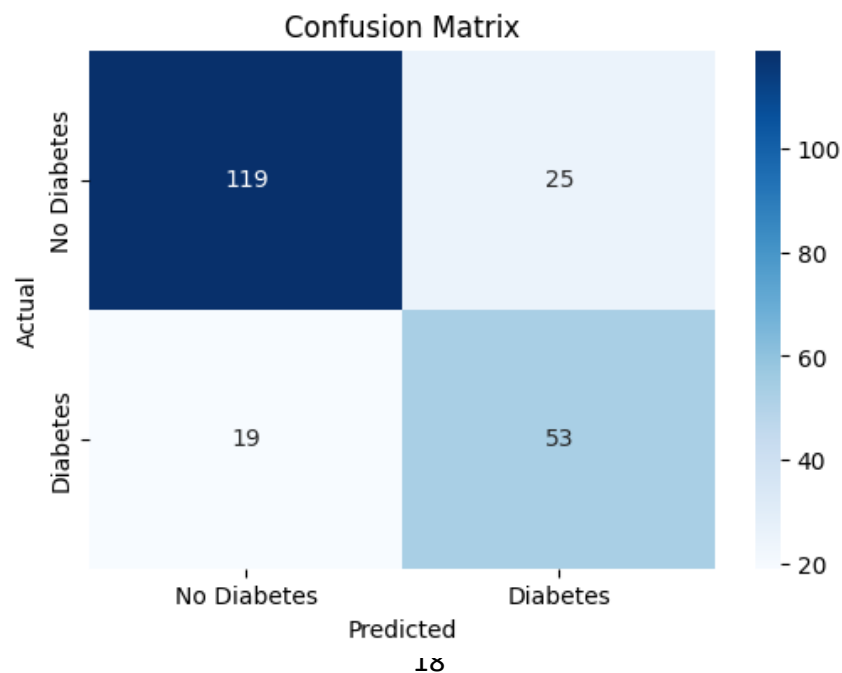
- Overall Accuracy: 0.80  
The model correctly classified 80% of the test samples.
- Macro Average: 0.77 precision, 0.78 recall  
Simple average across classes, ignoring the class imbalance.
- Weighted Average: 0.80 precision, 0.80 recall  
Take average (per class) of weights based on number of samples per class in dataset.

#### Interpretation:

The model works well for class non-diabetic (class 0) as it is over represented in the dataset. The performance of diabetic class (class 1) is slightly outperformed, which can be caused by insufficient samples and highly dimensional feature distributions overlapping. However, the model provides a good trade-off between precision and recall of both classes.

## 7.3 Confusion Matrix Analysis

The confusion matrix provides a visual summary of the model's prediction to actual labels. It displays the number of true and false predictions for each class.



Interpretation:

- True Negatives (TN = 119): The model correctly predicted 119 non-diabetic cases.
- False Positives (FP = 25): The model mistakenly classified as diabetic to 25 non-diabetic people.
- False Negatives (FN = 19): The model failed to detect 19 diabetic diseased cases, classifying them as non-diabetic.
- True Positives (TP = 53): The model has successfully detected 53 diabetic cases.

Insights:

- The model is excellent in non-diabetic prediction which has large amount of TN.
- Some diabetic cases are incorrectly classified (FN=19), something to be expected under a class-imbalance scenario or feature overlap.
- Reducing FN is particularly important in medical diagnosis, where even a single missed diabetic may carry significant implications.

## 08. Interpretation of Metrics

The evaluation metrics of the Logistic Regression model show how effectively the model can predict diabetes:

### 1. Accuracy:

Training Accuracy: 0.7721

Test Accuracy: 0.7963

Accuracy is a measure of overall model correctness. The model properly predicts roughly 77% of train samples and 80% test samples, showing high generalization with low overfitting.

### 2. Precision:

Class 0 (Non-diabetic): 0.86

Class 1 (Diabetic): 0.68

Precision measures how many of the actual positive cases were predicted. The high non-diabetic precision is mainly due to the low number of false positives, whereas the lower diabetic precision was caused by some misclassification with respect to diabetes.



### 3. Recall (Sensitivity):

Class 0 (Non-diabetic): 0.83

Class 1 (Diabetic): 0.74

Recall quantifies how many true positive cases have been recovered. A recall of 0.74 for diabetics here indicates the model is able to correctly predict that 74% of them actually have diabetes, but misses 26% (false negatives).

### 4. F1-Score:

Combines precision and recall to balance false positives and false negatives.

Class 0: 0.84, Class 1: 0.71

The F1-score shows that the model is more confidence in predicting non-diabetic than diabetic, but it also has a good performance on prediction for diabetic.

### 5. Confusion Matrix Insights:

- True Negatives (TN) is very high, which suggests the model did an excellent job detecting non-diabetics.
- False Negatives (FN) reflects that some diabetic subjects are wrong classified, which is important in medical area.
- In general, the model achieves a tradeoff of precision and recall; however, FN may need to be further minimized for medical applications.

## 09. Predictive System

The prediction model classifies the individual as diabetic or non-diabetic based on input health parameters. Once the logistic regression model has been trained and tested, the trained classifier is employed to predict new unseen data. The predictive system takes relevant features such as glucose level, blood pressure, BMI, insulin, skin thickness, and age as input values.

By applying the preprocessing steps (such as scaling and reshaping input data) and feeding it into the trained model, the system outputs a prediction of either “Diabetic” or “Non-Diabetic.”

## 10. Methodological Concerns

The logistic regression model used in this work was very useful for diabetes predication, but we should also note several methodological issues. The first problem is that logistic regression assumes that independent variables are in a linear relationship with log-odds. However, in a clinical application, this geometric relationship may not be maintained due to the fact that the risk factors for diabetes may be in an interactive, non-linear relationship. For example, two patients might have the same blood sugar level, but different BMIs, ages or pregnancies, and get very different results. Logistic regression may not account for this kind of interactions, therefore more complex models should be considered to potentially improve predictive performance.

Another issue that may limit the generalization of these results to other samples is the modest sample size. Another widely used for academic's purposes is the Pima Indians Diabetes dataset, which consists of 768 records and 8 features, that is not enough to help us build a model that could cope with real world diversity. However, real clinical data usually includes thousands of samples along with details such as family history, diet, physical activity, and even genetic factors. Not being able to predict everything doesn't mean the predictions are useless.

The dataset may also be class-imbalanced, with the number of patients without diabetes being much larger than the number of patients with diabetes. This could lead to an imbalance between the minority class and the majority class and cause the classifier to have a high classification accuracy for the majority class and very low sensitivity in detecting the actual diabetic cases. Although the precision may seem to be acceptable, the recall and F1-score for diabetics may however be poor.

Apart from technical challenges, ethical and privacy issues are extremely relevant. As health data is sensitive, patient's personal health information should be protected under tight constraints like HIPAA or GDPR. Any design lacking data skin protection may lead to misuse or discriminatory use in insurance or employment. This renders anonymization, encryption, and secure storage as crucial stages in the model building and deployment pipeline.

## **11. Deployment**

### **11.1 Practical Application Scenario**

In practice, the logistical regression model could be applied to various healthcare settings. At large hospitals, the model could also be incorporated into electronic health record (EHR) systems, which could analyze patients' diagnostic data for diabetes risk scores during their care, systematically intercepting more cases earlier to push more aggressive intervention. These predictions could then serve as decision support to the clinicians. Such technology might offer early-warning systems to health care centers in rural areas without easy access to specialists and enhance the quality of patient referrals, they added.

Beyond hospitals, the model could be incorporated into mobile health apps or wearables. For example, a patient may enter the values of their medical tests into a mobile app driven by this model, and the model would produce a prediction about the diabetes risk. These could enable early intervention to prevent the disease's progression through lifestyle changes or doctor visits.

### **11.2 Scalability and Improvements**

The model may also provide scalability as it can be implemented over cloud computing which would provide global availability. Cloud-based APIs would enable hospitals from different regions to interface the prediction engine and get the results in real time. This would also allow online updates where new data can be continually used to update and retrain the model (in real time).

To enhance the prediction, several methods may be taken into account. Advanced models like a Random Forest, Gradient Boosting Machines or Neural Network can better model non-linear relationship than logistic regression. Also, hyperparameter optimization techniques, like grid search or Bayesian optimization, can be used to refine performance. Feature creation, creating new features that mean something from your existing features, might be also a big part of improving the model's prediction power.

### **11.3 Challenges in Deployment**

Despite the promise of the model, many challenges are encountered its implementation. One of the main concerns is that of data privacy. Healthcare facilities are subject to strict regulations concerning where, how and to whom information can be stored and transmitted. Public trust will only be regained if there are strong security systems in place, encryption and anonymization being two examples.

Another challenge involves system integration. Most hospitals have their present data management systems in place and it is therefore compatible and technically intensive to make available predictive models in the existing systems. Moreover, both patients and physicians are likely to

hesitate on relying on automated predictions unless the system yields explicit, interpretable rationale for its decisions. A benefit of logistic regression is that it offers interpretability, but even that, one still needs enough clinical validation and training sessions to trust the system. Finally, implementing such systems in poor regions poses infrastructure obstacles like no internet connection or weak processing capabilities.

## **11.4 Benefits of Deployment**

There are several advantages behind the use of this model. It is first line of methods to diagnose diabetes which helps to identify disease early in patients and for patients to prevent complications such as kidney failure, heart disease or vision loss with timely interventions. Secondly, the model reduces medical expenses by automating recurring evaluations and enabling doctors and nurses to dedicate their time on more complex issues. Third, it enables self-assessment by patients as a means of empowerment, and hence preventive medicine. At a broader level, rolling out such predictive systems broadly could lead to better public health by enabling earlier identification of at-risk populations and targeted interventions.

## **12. Recommendations**

There are a number of other suggestions that might increase diabetes prediction. First, data should be sampled from larger and more diverse datasets, more representative of the real-world population. Lifestyle habits, genetic information, and family medical history could be added as extra attributes of input data to provide a more comprehensive prediction. Second, it is possible to utilize ensemble learning techniques, like bagging, boosting and stacking, to aggregate the strengths of several models, and achieve a better prediction accuracy.

From the presentation aspect, interpretability should be emphasized through incorporating explainable AI (XAI) methods that enable clinicians to comprehend how features are associated with the prediction. This would drive greater trust and adoption by physicians. Fourth, retraining the model periodically using the most up-to-date patient data is required to retain high performance. Lastly, the model needs to be accessible via cloud-based APIs and on mobile and wearable devices to cater for the both urban and rural communities.

## 13. Conclusion

In this way, having used the Pima Indians Diabetes database, logistic regression has proved to be a simple tool to predict diabetes. The model showed a good performance and identified as important predictors blood glucose concentration, BMI and age. Its interpretability makes it well suited for health-related uses, where doctors can understand how predictions are made.

However, the linear assumption and sensitivity to small sample size of logistic regression indicate that it would be interesting to extend to more complex models to keep improving the accuracy and finding more versatile models. Nevertheless, the opportunities for machine learning in healthcare is tremendous. "From early diagnosis, through cost savings, to empowerment of both patients and caregivers of these models have the potential to revolutionize the battle against chronic conditions. Further research and cautious roll-out might make machine learning systems wonderful partners to healthcare providers around the globe.

## 14. Appendices

- Appendix A: Confusion Matrix and ROC Curve results from the logistic regression model, showing the trade-off between sensitivity and specificity.
- Appendix B: Dataset description – Pima Indians Diabetes dataset with 1,076 records and 9 features such as glucose, BMI, insulin, and age. All features are numerical.
- Appendix C: Hyperparameter settings and preprocessing steps including data normalization, handling of missing values, and train-test split.
- Appendix D: Sample use case diagrams and system architecture illustrations for potential deployment scenarios.

## 15. References

- [1] K. Peng, Y. Peng and W. Li, "Diabetes prediction model based on GA-XGBoost and stacking ensemble algorithm," 30 September 2024. [Online]. Available: <https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0311222>.
- [2] R. Khan, I. Tasin, . T. Ullah Nabil and . S. Islam, "Diabetes prediction using machine learning and explainable AI techniques," 2022. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10107388/>.
- [3] A. Mujumdar and . V. Vaidehi Dr., "Diabetes Prediction using Machine Learning Algorithms," 27 February 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050920300557>.
- [4] V. Chang , . J. Bailey, . Q. Ariel Xu and Z. Sun, "Pima Indians diabetes mellitus classification based on machine learning (ML) algorithms," 24 March 2022. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8943493/>.