** A**

**Assesment Report**

on

**“Problem Statement”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**Name of discipline**

By

Name : SHIVAM KUMAR

Roll No : 12

**Under the supervision of**

“ABHISHEK SHUKLA”

**KIET Group of Institutions, Ghaziabad**

Affiliated to

**Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

(Formerly UPTU)

**May, 2025**

# INTRODUCTION :

1. **Binary Classification Problem**
   * The target variable is Outcome, which has only two values:
     + 0 = The patient **does not** have diabetes
     + 1 = The patient **does** have diabetes
   * Our task is to build a machine learning model that can **predict this outcome** based on the input features.

# Missing or Unrealistic Values

* + Some features like Glucose, BloodPressure, SkinThickness, Insulin, and BMI may contain zeros, which are **medically impossible** and should be treated as **missing values**.

1. **Class Imbalance**
   * The number of non-diabetic patients (Outcome = 0) is **greater** than diabetic ones (Outcome = 1), which can cause a model to be biased toward predicting non-diabetic outcomes unless handled carefully.
2. **Correlated Features**
   * Some features may be correlated (e.g., Glucose and Insulin), which might affect model performance and need to be analyzed using correlation matrices.
3. **Feature Scaling**
   * The features are on **different scales**, so scaling (e.g., with StandardScaler) is essential for many ML algorithms to perform correctly.

# METHODOLOGY:

To address the diabetes prediction problem presented in the dataset, we followed a systematic machine learning workflow consisting of several stages — from data preprocessing to model evaluation. The entire approach is outlined below:

**1. 📥 Data Loading and Exploration**

* The dataset was loaded using **Pandas** to examine the structure, check for missing or invalid values, and understand the distribution of features.
* Basic statistical analysis and data visualizations (e.g., bar plots, histograms, heatmaps) were performed to get insights into feature distributions and correlations.

**2. 🧹 Data Cleaning and Preprocessing**

* **Missing and Invalid Values**: Zero values in features like Glucose, BloodPressure, BMI, SkinThickness, and Insulin were identified as biologically implausible. These were either:
  + Replaced using imputation techniques (e.g., mean/median imputation), or
  + Rows with these values were dropped for certain models depending on the preprocessing strategy.
* **Feature Scaling**: Since the features had different units and scales, **StandardScaler** from scikit-learn was used to normalize the data. This step is essential for distance-based models and helps speed up model convergence.
* **Class Balance**: The target variable (Outcome) was imbalanced. Techniques like stratified sampling during the train-test split were used to maintain the same class distribution across training and testing sets.

**3. ✂️ Train-Test Split**

* The data was split into **training (80%) and testing (20%)** subsets using scikit-learn’s train\_test\_split function, with a fixed random seed for reproducibility.

**4. 🤖 Model Selection and Training**

* A **Random Forest Classifier** was chosen as the primary model due to its:
  + High accuracy
  + Built-in handling of feature interactions
  + Resistance to overfitting
  + Ability to rank feature importance
* The model was trained using the scaled training data.

**5. 📈 Model Evaluation**

* The model’s performance was assessed using various metrics on the test data:
  + **Accuracy**: Overall correctness of the model
  + **Confusion Matrix**: Visualization of true vs predicted classes
  + **Precision, Recall, and F1-Score**: To evaluate performance on each class (diabetic and non-diabetic)
* These metrics provide insights into not just how often the model is correct, but also how well it handles false positives and false negatives — which are critical in medical diagnoses.

**6. 📊 Data Visualization**

* Several plots were generated to support the analysis:
  + **Outcome Distribution**: Bar chart of class frequencies
  + **Correlation Heatmap**: Understanding feature relationships
  + **Pairplots**: Visualizing class separation among key features like Glucose, BMI, and Age

# CODE:

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix, classification\_report

# Load dataset

df = pd.read\_csv("2. Diagnose Diabetes.csv")

# ------------------ Data Visualization ------------------

# Bar plot of Outcome distribution

sns.countplot(x='Outcome', data=df, palette='Set2')

plt.title('Distribution of Diabetes Outcome')

plt.xlabel('Outcome (0 = No, 1 = Yes)')

plt.ylabel('Count')

plt.show()

# Heatmap of correlation between features

plt.figure(figsize=(10, 8))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Feature Correlation Heatmap')

plt.show()

# Clean data for pairplot (ensure numeric types)

cols = ['Glucose', 'BMI', 'Age', 'Outcome']

for col in cols:

    df[col] = pd.to\_numeric(df[col], errors='coerce')

df\_clean = df[cols].dropna()

# OUTPUT:

A graph of a number of diabetes outcomes

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.