Module Code & Module Title
CU6051NI Artificial Intelligence

75% Individual Coursework
Submission: Final Submission
Academic Semester: Autumn Semester 2025
Credit: 15 credit semester long module

Student Name: Kribip Thapa
London Met ID: 23056252
College ID: np01cp4s240118@islingtoncollege.edu.np
Assignment Due Date: 21st January, Wednesday
Assignment Submission Date: 21st January, Wednesday
Submitted To: Er. Mukesh Regmi

GitHub Link	

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

1. Introduction

1.1 Explanation of the Chosen Topic

The topic chosen for this assignment is Diabetes risk prediction using the population level health indicators. Diabetes is a life-long health issue which affects the body ability to maintain the blood glucose levels and has also come to become a very crucial public health topic. The day to day increasing patients of diabetes is directly related to the lifestyle changes, urbanization, physical inactivity, heredity and dietary habits.

Rather than mostly focusing on the clinical laboratory tests, the current topic displays the analysis of health indicators and lifestyle factors which is mostly affected by public health surveys. These kinds of indicators include the body mass index, hypertension, physical inactivity, smoking behaviors and general health conditions. All these analysis shows that the day-to-day behaviors and health states affects the possibility of developing diabetes over time.

The selection of this topic is due to support the early risk identification at a mass level. By thoroughly checking and analyzing patterns within a huge scale health indicator data it makes it bit easier to identify individuals and the groups which may be at the risk of diabetes before some severe symptoms appear. It is very useful for preventive healthcare plannings, health awareness programmes and decision support system mainly focused at reducing long term effects of the diabetes on individuals and healthcare systems.

1.2 Explanation of the AI Concepts Used

This assignment includes the fundamental concepts from Artificial Intelligence (AI), which a focus on Machine Learning (ML). AI means the design of systems that can perform works which requires human like intelligence, such as recognizing patterns, learning from the data and making predictions. ML is a important subfield of AI which lets systems to

automatically learn relationships from data without being explicitly programmed, which makes it well fit for a data driven decision making problems.

The research mainly focuses on supervised machine learning, where the AI models are trained using some datasets which consist of both the input and output labels. After this approach, the algorithms learn the relationship between health-related issues and the similar outcome. The supervised learning task discussed in this assignment is binary classification, which includes giving each data instance to one of two possible categories.

In this topic, the binary classes whether an individual is diabetic or non-diabetic. Supervised classification is right for this issue because the dataset contains real health records with known outcomes, which lets effective training and evaluation of the models. Similar supervised algorithms including K-Nearest Neighbors (KNN), Naïve Bayes and Decision Tree were chosen for this assignment because of their ability to handle the structured data and provide predictive results.

1.3 Explanation / Introduction of the Chosen Problem Domain

Global Scenario

Diabetes is recognized as a vital challenge for public health systems across the world due to its long-term impact and high prevalence. International health organization reports a increasing number of people affected by diabetes, affected directly by their modern lifestyle patterns rather than sudden medical causes. Things such as low physical activity, increased consumption of calorie-dense foods and the population aging have also contributed to this trend.

According to the global health estimates the World Health Organization (WHO) the adults living with diabetes has grown over recent decades, reaching hundreds of millions worldwide. This increase has transformed diabetes from individual health issue to large

scale societal concern. One of the main challenges related with diabetes is that it often progresses silently, meaning (World Health Organization, 2023)

International Diabetes Federation (IDF) has calculated that the global burden of diabetes will continue to arise in the coming years, particularly in regions where healthcare access and routine screening are very limited. Such estimation shows the need for approaches which moves beyond late-stage diagnosis and focus instead of identifying risk patterns early. From a data perspective, it makes diabetes a suitable domain for predictive analysis as its risk is strongly connected with measurable lifestyle and health related indicators that can be captured through large population surveys. (International Diabetes Federation, 2023)

Nepal Scenario

Within Nepal, diabetes has become an increasingly important public health concern, especially in urban and rapidly developing areas. Changes in daily routines, dietary habits, and occupational patterns have altered the health profile of large sections of the population. National-level health assessments and reports from organizations such as the **(**Nepal Health Research Council**)** and the WHO indicate that diabetes prevalence is rising alongside other non-communicable conditions.

Evidence from the WHO stepwise Survey conducted in Nepal suggests that diabetes risk is closely linked with indicators such as body weight, physical inactivity, smoking behavior, and high blood pressure. These indicators are not isolated medical measurements but reflect broader lifestyle trends that are becoming more common within the country. The survey also highlights differences in prevalence between urban and rural populations, suggesting that environmental and behavioral factors play a significant role.
(World Health Organization, 2023)

Nepal faces practical challenges in managing long-term health conditions due to limited healthcare resources and uneven access to diagnostic services, particularly in rural regions. As a result, preventive strategies that rely on early identification of risk are especially valuable. Analyzing diabetes risk using health indicators offers a cost-effective and scalable approach that aligns with Nepal's public health needs.

Although the dataset used in this coursework is based on international public health data, the risk factors represented such as physical inactivity, obesity, smoking, and general health status are highly applicable to the Nepali context. Therefore, the problem domain of diabetes risk prediction using health indicators is both relevant and transferable, supporting the development of data-driven insights that can inform future health planning and awareness initiatives in Nepal. (Nepal Health Research Council, 2023)

1.4 Relevance of the Problem Domain to This Coursework

The problem domain of diabetes risk prediction utilizing health and lifestyle indicators connects with the objective of this coursework. It shows a practical, real-world scenario where large and structured datasets are available, making it easy for the application of supervised machine learning techniques. The use of non-clinical health indicators lets the meaningful analysis without depending on intensely hardcore medical data, making sure the problem remains appropriate for an academic machine learning study.

By approaching diabetes as a risk prediction task rather than a medical diagnosis, this assignment mainly focuses on the importance of data driven pattern recognition in preventive healthcare. This framing not only displays the practical relevance of machine learning in addressing real societal challenges but also provides a technically suitable context for exploring and evaluating classification algorithms within an educational setting.

1.5 Scope and Objective of This Coursework

The main objective of this assignment is to apply supervised machine learning techniques to population level health to predict diabetes risk. The study focuses on identifying similar health and lifestyle indicators, developing a structured machine learning solution, and implementing multiple classification algorithms to evaluate their predictive performance. Through practical model training, testing and analysis, the assignment shows how machine learning can be effectively used to support data driven insights in the context of preventive healthcare.

2. Background

2.1 Research Work on Diabetes Risk Prediction

2.1.1 Diabetes Risk Prediction as a Research Problem

Diabetes risk prediction has become an important research area within public health, data science, and artificial intelligence due to the long-term and progressive nature of the disease. Unlike acute medical conditions, diabetes develops gradually and is strongly influenced by lifestyle, behavioral, and demographic factors such as physical inactivity, obesity, smoking habits, dietary patterns, and general health conditions. Because symptoms may not be immediately visible, a large proportion of individuals remain undiagnosed until complications arise.

Previous research highlights that early identification of individuals at high risk of diabetes can support preventive healthcare strategies, reduce long-term treatment costs, and minimize complications such as cardiovascular disease, kidney failure, and vision loss. As a result, researchers have increasingly explored predictive approaches that rely on existing health-related data rather than laboratory-based diagnostic tests alone. Consequently, diabetes risk prediction is widely studied as a classification problem, where individuals are categorized into diabetic or non-diabetic risk groups based on observed indicators. (World Health Organization, 2023)

2.1.2 Application of Machine Learning in Diabetes Risk Prediction

Machine learning has been widely adopted in diabetes risk prediction research because of its ability to analyze multiple health indicators simultaneously and detect complex patterns within large datasets. Traditional statistical methods often rely on strong assumptions and limited variable interactions, whereas machine learning models can automatically learn relationships between features and outcomes.

Most studies in this domain apply supervised learning techniques**,** where labelled datasets are used to train models to predict diabetes status. Commonly used algorithms include Logistic Regression, Decision Trees, Random Forests, Support Vector Machines, Naïve Bayes, and K-Nearest Neighbors. Research findings indicate that ensemble models such as Random Forest often achieve higher predictive accuracy, while simpler models such as Logistic Regression offer better interpretability, which is particularly important in healthcare-related applications. (International Diabetes Federation, 2023)

2.1.3 Role of Health Indicators and Lifestyle Factors in Existing Studies

A major trend in diabetes prediction research is the use of **non-invasive health indicators** instead of relying solely on clinical laboratory tests. Studies frequently utilise indicators such as body mass index (BMI), age group, blood pressure status, cholesterol levels, smoking behavior, alcohol consumption, physical activity, mental health indicators, and general health status.

These variables are commonly collected through national health surveys and administrative datasets, making them accessible, scalable, and cost-effective for large population studies. Existing research suggests that incorporating lifestyle and behavioral factors significantly improves early risk identification because diabetes development is influenced by long-term habits rather than isolated clinical measurements. Despite

limitations related to self-reported data, survey-based indicators remain widely used due to their relevance to preventive healthcare planning. (Centers for Disease Control and Prevention, 2023)

2.2 Review and Analysis of Existing Work in the Problem Domain

2.2.1 Population-Based Studies Using Health Survey Data

Many existing studies have utilized large-scale public health survey datasets to predict diabetes risk at a population level. Research accessed through academic platforms such as Google Scholar and PubMed displays the utilization of national survey data containing health and lifestyle indicators to train supervised machine learning models. These studies focus on identifying high-risk individuals rather than performing clinical diagnosis.

Results from population-based studies indicate that indicators such as BMI, age, smoking status, physical activity, and blood pressure provide meaningful predictive signals. However, researchers also acknowledge limitations related to self-reported values, which may introduce bias or measurement errors. Despite these challenges, survey-based approaches remain popular due to their scalability and applicability in public health decision-making.

2.2.2 Studies Based on Clinical and Administrative Health Records

In contrast, some studies rely on clinical and administrative health records obtained from hospitals, insurance systems, or electronic health records. These datasets often include laboratory measurements and longitudinal patient information, enabling researchers to predict diabetes onset over extended time periods.

While models trained on clinical data often achieve higher predictive accuracy, they require access to sensitive medical records and advanced healthcare infrastructure. This

limits their use in low-resource settings and large-scale screening programmes. Additionally, the reliance on laboratory tests increases cost and complexity, making such approaches less suitable for preventive and population-wide analysis.

2.3 Analysis of the Existing Scenario Using Different Algorithms

2.3.1 K-Nearest Neighbors (KNN) Algorithm 1 Problem Statement

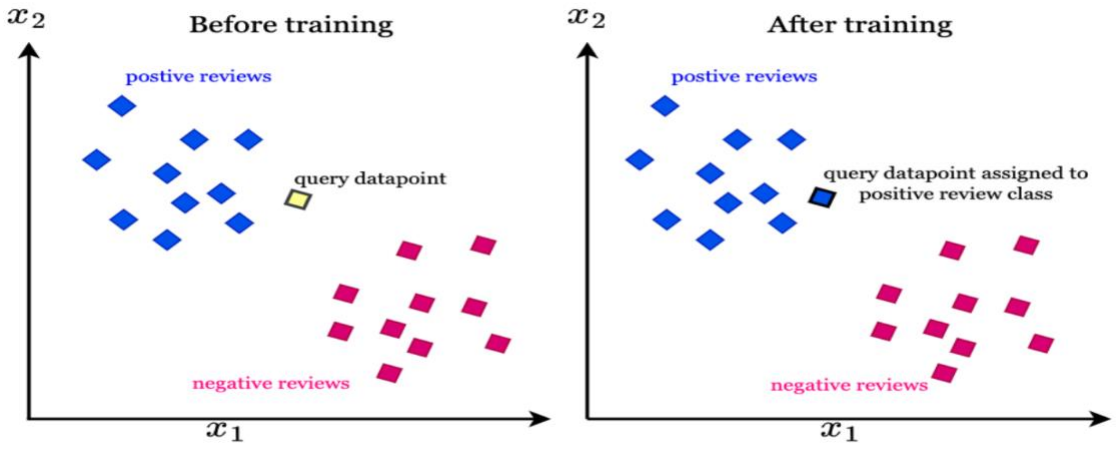

Figure 1 KNN

Numerous research uses the K-Nearest Neighbors (KNN) algorithm as a supervised classification method for diabetes risk prediction. Health indicators including BMI, age group, blood pressure, smoking habits, and physical activity are used in this method to represent individuals. By comparing a person's health profile to similar cases in the dataset, KNN assigns the most common class among the closest neighbors.

According to research, KNN is easy to use and doesn't require complicated model training, and it works well on structured health datasets. Its primary drawbacks, however,

are its sensitivity to feature scaling, the selection of k, and higher processing costs when working with big datasets.

2.3.2 Problem Statement Using Algorithm 2: Decision Tree

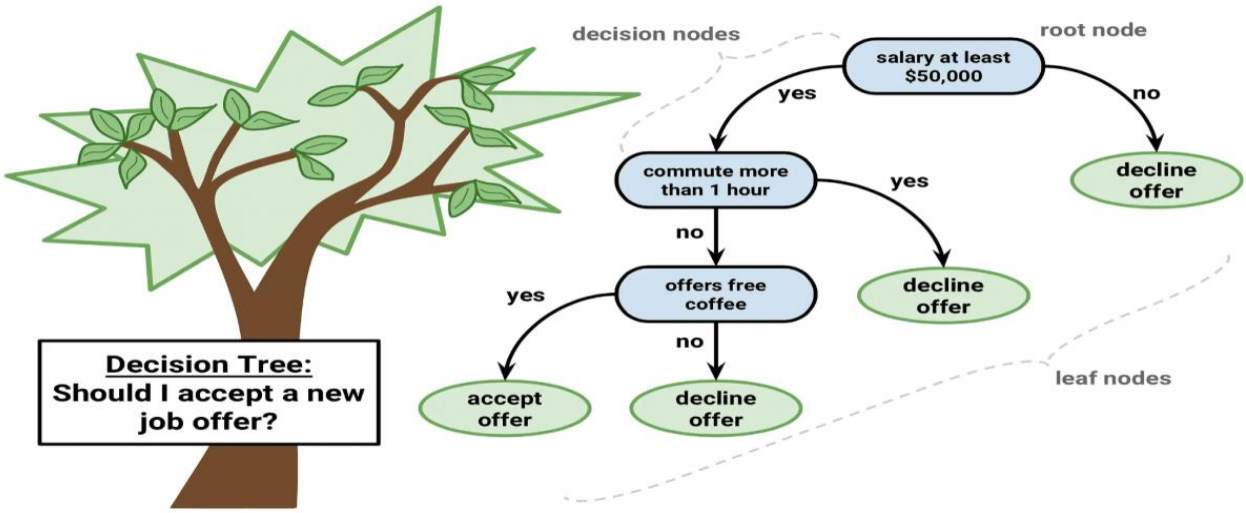

Figure 2 Decision Tree

Decision Tree models is most used to capture non-linear relationships between health indicators in classification tasks. Decision Tree work by recursively dividing dataset based on decision rules derived from input features, allowing them to show meaningful combinations of risk factors, like high body mass index combined with low physical activity, because of their hierarchical structure, Decision Trees are easy to interpret and provide clear insights into how different health attributes contribute to diabetes risk. Its ability to handle structured data makes it most suitable to survey based health datasets used in this coursework.

2.3.3 Problem Statement Using Algorithm N: Naïve Bayes

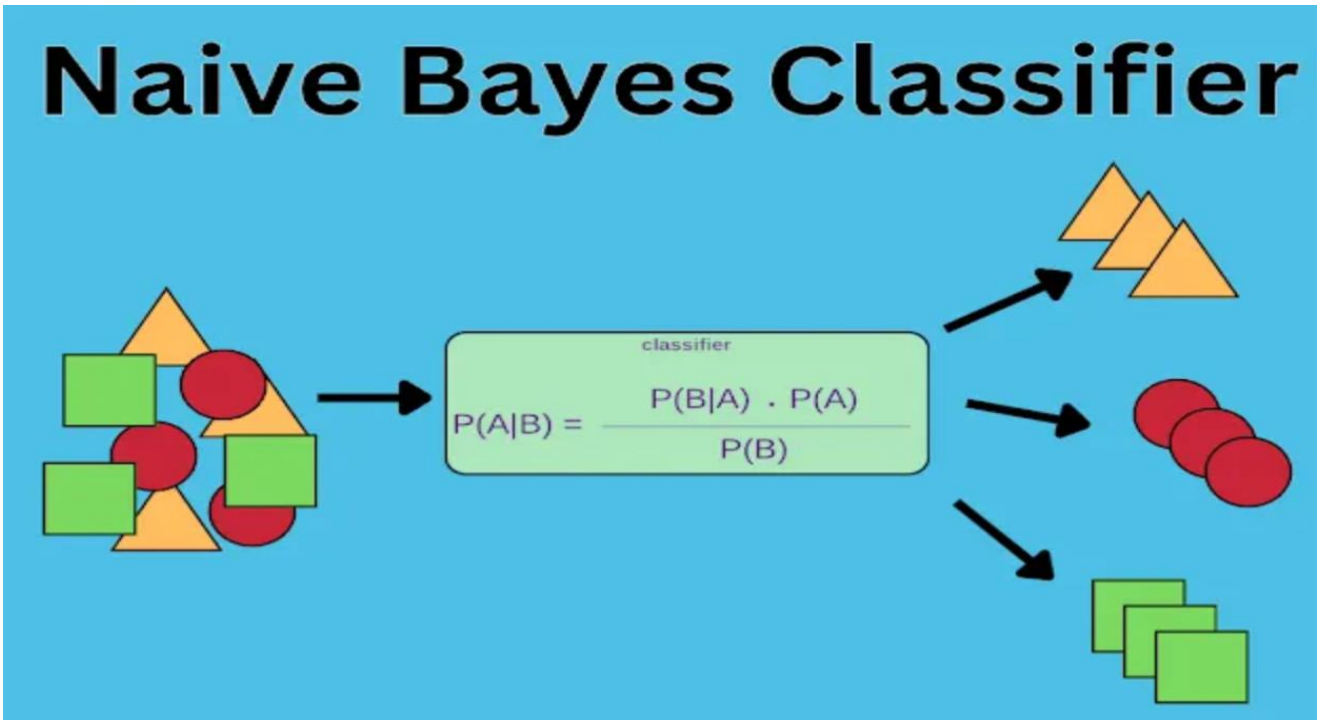

Figure 3 Naive Bayes

Some studies apply algorithms such as K-Nearest Neighbors (KNN) or Naïve Bayes for diabetes risk prediction. KNN classifies individuals based on similarity to other cases, while Naïve Bayes applies probabilistic assumptions to estimate diabetes likelihood.

These algorithms are computationally efficient and simple to implement but may struggle with high-dimensional data or correlated features. Research often uses them for comparative analysis rather than final deployment due to performance sensitivity to parameter selection and data distribution.

2.3.4 Summarised Review and Analysis

Overall, existing research demonstrates that diabetes risk prediction is effectively addressed using supervised machine learning techniques. Logistic Regression is commonly used for interpretability, tree-based models for improved accuracy, and probabilistic or distance-based models for comparative evaluation. However, recurring challenges include data imbalance, self-reported bias, limited interpretability, and reduced generalizability across populations. These findings highlight the need for balanced predictive approaches that combine performance, transparency, and real-world applicability.

2.4 Dataset Information and Dataset Background

The dataset used in this coursework is derived from a large-scale public health survey focusing on diabetes-related health indicators. It contains over 190,000 records and includes demographic, lifestyle, and health-related variables with a binary target label indicating diabetes status.

Key Dataset Attributes Include:

- **Demographic:** Age group, Sex, Education level, Income
- **Health Indicators:** BMI, High Blood Pressure, High Cholesterol, General Health
- **Lifestyle Factors:** Physical Activity, Smoking, Alcohol Consumption, Fruit and Vegetable Intake
- **Medical History:** Heart Disease, Stroke
- **Target Variable:** Diabetes binary (Diabetic / Non-Diabetic)

This dataset is suitable for supervised classification tasks due to its size, structured format, and relevance to population-level risk analysis. Its focus on lifestyle and health

indicators aligns with preventive healthcare objectives and allows the application of interpretable machine learning models.

3. Solution

3.1 Explanation of the Proposed Solution

3.1.1 Proposed Solution

The proposed solution prioritizes on developing a machine learning based diabetes risk prediction system using population level and lifestyle indicators. The system is built to recognize individuals who are at higher risk of developing diabetes at an early stage, rather than performing a clinical diagnosis. For analyzing structured health data gathered through public health surveys, the solution helps preventive healthcare decision making and awareness.

The solution handles diabetes risk prediction as a binary classification problem, where individuals are grouped as Diabetic or Non-Diabetic. Indicators such as age, BMI, blood pressure status, cholesterol levels, physical activity, smoking behavior, general health condition are utilized as an input features. These shows long term lifestyle and health patterns which is strongly connected with diabetes risk.

3.1.2 Approach to Solving the Problem

The strategy follows a structured machine learning workflow. First of all, dataset is loaded and processed to eradicate pointless attributes, handle missing values, and prepares features for the evaluation. The final cleaned data is parted into training and testing subsets to make sure neutral review.

Numerous supervised learning algorithms are prepared on the dataset to know relationships between health indicators and diabetes status. Training more than one model allows correlation of performance and durability. Every model is evaluated using

metrics such as accuracy precision, recall, F1-score, also confusion matric to make sure even-handed assessment.

The best-performing and most interpretable model is selected as the final solution. This trained model can then be used to predict diabetes risk for new individuals based on their health indicator data, providing a practical and scalable tool for early risk identification aligned with the objectives of this coursework.

3.2 Explanation of the AI Algorithms Used

3.2.1 AI Algorithms Used in This Project

This falls under supervised machine learning algorithm to implement the issue of diabetes risk prediction as a binary classification task, where individuals are categorized as diabetic or non-diabetic. According to the nature of the dataset, which contains of structured health and lifestyle indicators, three well established classification algorithms are selected:

1. K-Nearest Neighbours (KNN)

2. Naïve Bayes

3. Decision Tree

These algorithms are used to work well in healthcare. These algorithms are fast, accurate and easy to understand. By testing many at once, we can pick the best for the work.

3.2.2 K-Nearest Neighbours (KNN)

K-Nearest Neighbors (KNN) is a distance based supervised learning algorithm that labels a data instance by checking the classes of its nearest neighbors in the feature space. This

project, KNN predicts diabetes level by testing an individual's health indicators such as age, BMI, physical activity, blood pressure and general health with same records in the dataset.

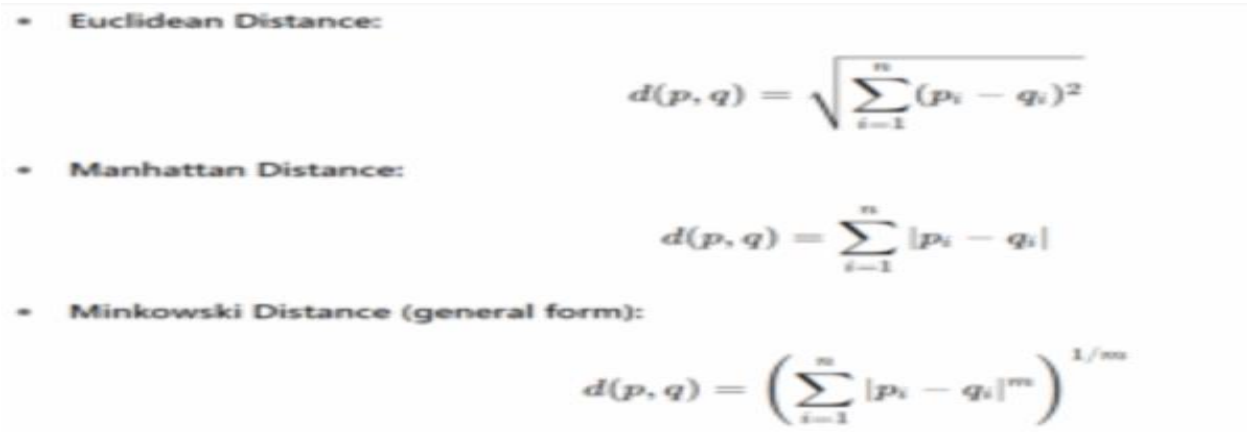

Figure 4 Forumla 1

Algorithm does not expect any raw data distribution, which makes it easy for real-world health data where relationships in between variables are often non-linear. Classification is known as majority vote among the k closest data points. Because KNN is dependent on distance calculations, scaling feature is crucial to avoid variables with huge numeric ranges from leading the classification process.

The most commonly used distance metric is Euclidean Distance, defined as:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_d - y_d)^2}$$

Other distance measures that can also be applied depending on data characteristics include:

- **Manhattan Distance**

n

$$DManhattan(x, y) = \sum_{i=1} | xi - yi |$$

- **Minkowski Distance (generalized form)**

$$D(x, y) = (\sum_{ni=1} | xi - yi |p)1/p$$

3.2.3 Naïve Bayes

Naïve Bayes is likely a supervised learning algorithm related to Bayes Theorem. It guesses class labels by figuring out the probability which is a given data instance belongs to a class based on observed feature values.

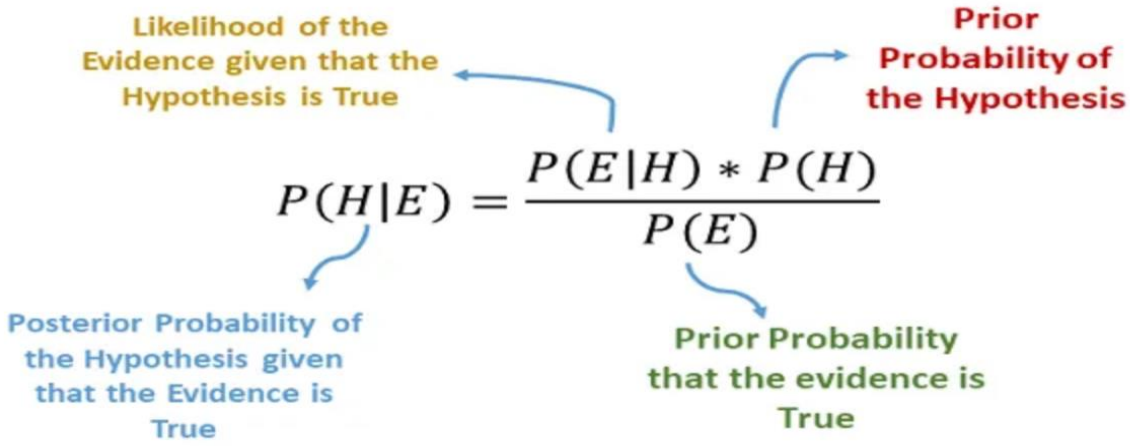

Figure 5 Formula 2

The algorithm expects that all features are solely independent, which eases computation and allows fast model training. Naïve Bayes guesses the likelihood of an individual being diabetic based on indicators such as, BMI cholesterol level, and general health status.

Bayes' Theorem is expressed as:

$$P(y \mid x) = \frac{P(x \mid y) \cdot P(y)}{P(x)}$$

For classification, the predicted class is determined as:

$$y_{new} = \arg\max_{y} P(y) \prod_{j=1}^{d} P(a_j \mid y)$$

Naïve Bayes performs well on structured datasets and is light however, its strong independence assumption may limit performance when features are highly connected, which is common in medical data.

3.2.4 Decision Tree

Decision Tree is a supervised learning algorithm that guesses findings by repeating splitting the dataset into smaller parts based on feature values. It builds a tree-like structure where each turning point shows a decision rule, and each end point shows a class label.

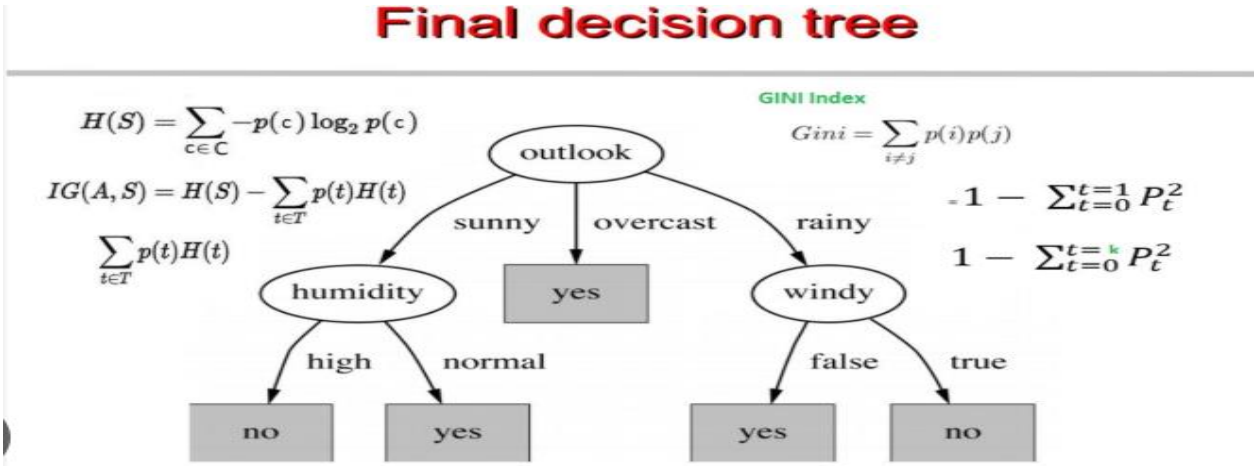

Figure 6 Formula 3

The Decision Tree picks out crucial health indicators such as BMI, age, blood pressure to decide diabetes risk. The model is mostly useful in healthcare applications because its decision logic is easy to interpret, allowing insights into how predictions are made.

Feature selection during splitting is guided by impurity measures such as:

- **Information Gain (IG)**

$$IG = H(parent) - \sum_{i} \frac{|\ child_i\ |}{|\ parent} H(child_i)$$

- **Gini Index**

$$Gini = 1 - \sum p_i^2$$

i=1

Even though Decision Trees are simple and clear, they are likely to overfitting if the tree becomes too hard. This problem is addressed through parameter tuning and evaluation on unseen test data.

3.3 Pseudocode

3.3.1 Pseudocode for K-Nearest Neighbours (KNN) Algorithm

START

Step 1: Load Dataset

LOAD diabetes health indicators dataset from CSV file

DISPLAY dataset to understand structure and attributes

#Step 2: Data Checking

CHECK for missing values

CONFIRM no missing values for present

 # Step 3: Data Preparation

ENCODE target variable (Diabetic/Non-Diabetic-numerical values)

REMOVE ID column

APPLY one-hote coding encoding to categorical features

#Step 4: Data Splitting

SPLIT dataset into training set and testing set

 # Step 5: Feature Scaling

APPLY feature scaling using Standard Scaler

FIT Scaler on training data and

TRANSFORM both training and Testing data

#Step 6: Model Initialization

SET number of neighbours(k=5)

INITIALIZE KNN classifier

#Step 7: Model Training

TRAIN KNN model using scaled training data

#Step 8: Prediction

PREDICT diabetes status using scaled testing data

 # Step 9: Evaluation

END

3.3.2 Pseudocode for Naïve Bayes Algorithm

START

 # Step 1: Load Dataset

LOAD diabetes health indicators dataset from CSV file

REVIEW dataset structure and target variable

 # Step 2: Data Checking

CHECK for missing values

CONFIRM no missing values are present

#Step 3: Data Preparation

ENCODE target variable into numerical form

REMOVE ID column

APPLY one-hot encoding to categorical features

#Step 4: Data Splitting

SPLIT dataset into training set and testing set

 # Step 5: Feature Scaling

APPLY feature scaling using StandardScaler

 # Step 6: Model Initialization

INITIALIZE Gaussian Naïve Bayes classifier

 # Step 7: Model Training

TRAIN Naïve Bayes model using training data

 # Step 8: Prediction

PREDICT diabetes status using testing data

#Step 9: Evaluation

CALCULATE accuracy of the Naïve Bayes model

END

3.3.3 Pseudocode for Decision Tree Algorithm

START

 # Step 1: Load Dataset

LOAD diabetes health indicators dataset from CSV file

EXAMINE dataset features and target label

 # Step 2: Data Checking

CHECK for missing values

CONFIRM no missing values are present

 # Step 3: Data Preparation

ENCODE target variable into numerical form

REMOVE ID column

APPLY one-hot encoding to categorical features

 # Step 4: Data Splitting

SPLIT dataset into training set and testing set

#Step 5: Model Initialization

SET maximum tree depth

INITIALIZE Decision Tree classifier

 # Step 6: Model Training

TRAIN Decision Tree model using training data

 # Step 7: Prediction

PREDICT diabetes outcome using testing data

 # Step 8: Evaluation

CALCULATE accuracy of the Decision Tree model

END

4.Flowchart

4.1 Flowchart of KNN algorithm:

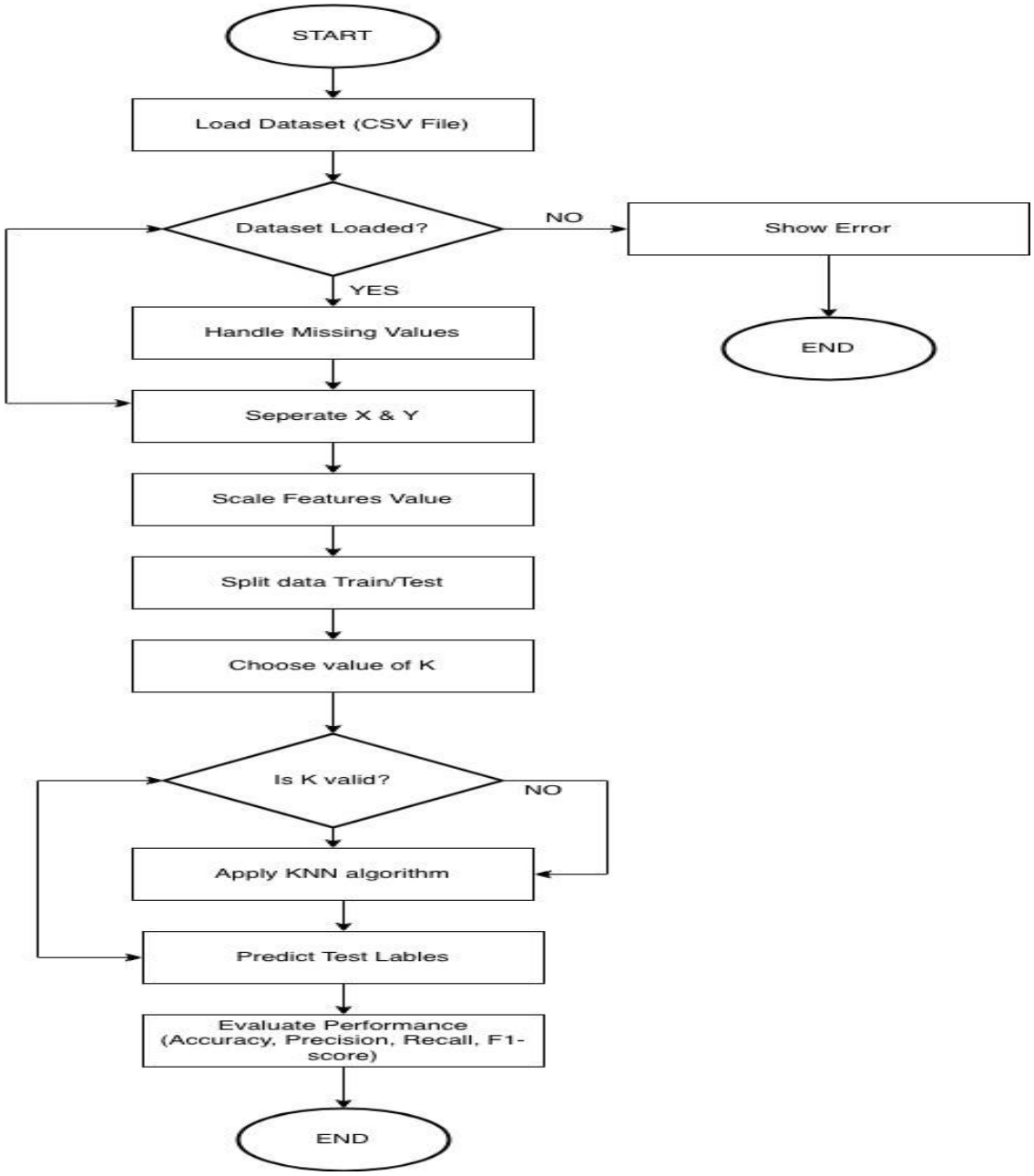

Figure 7 KNN Flowchart

4.2 Flowchart of decision tree algorithm

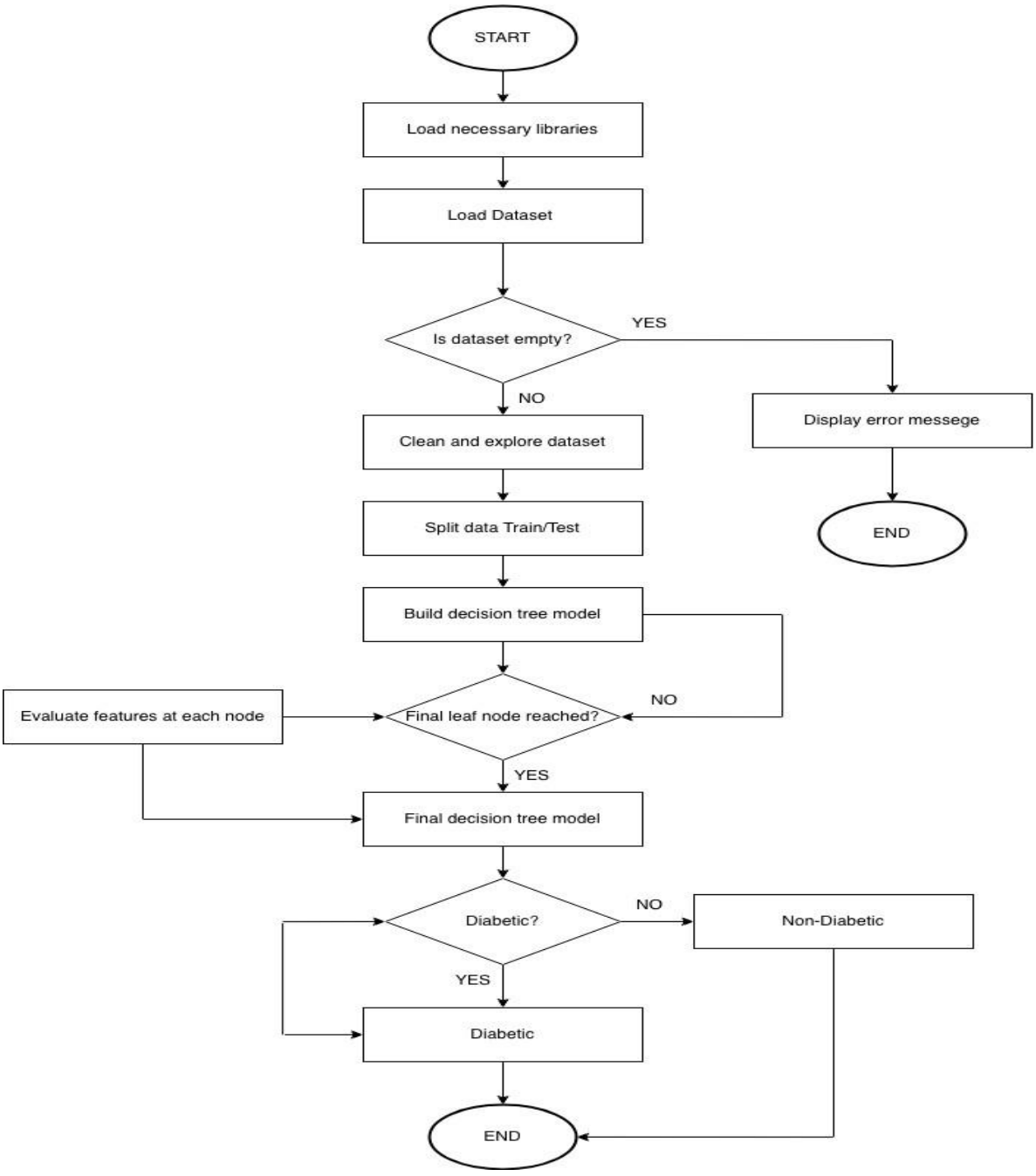

Figure 8 Decision Tree Flowchart

4.3 Flowchart of Naive bayes algorithm

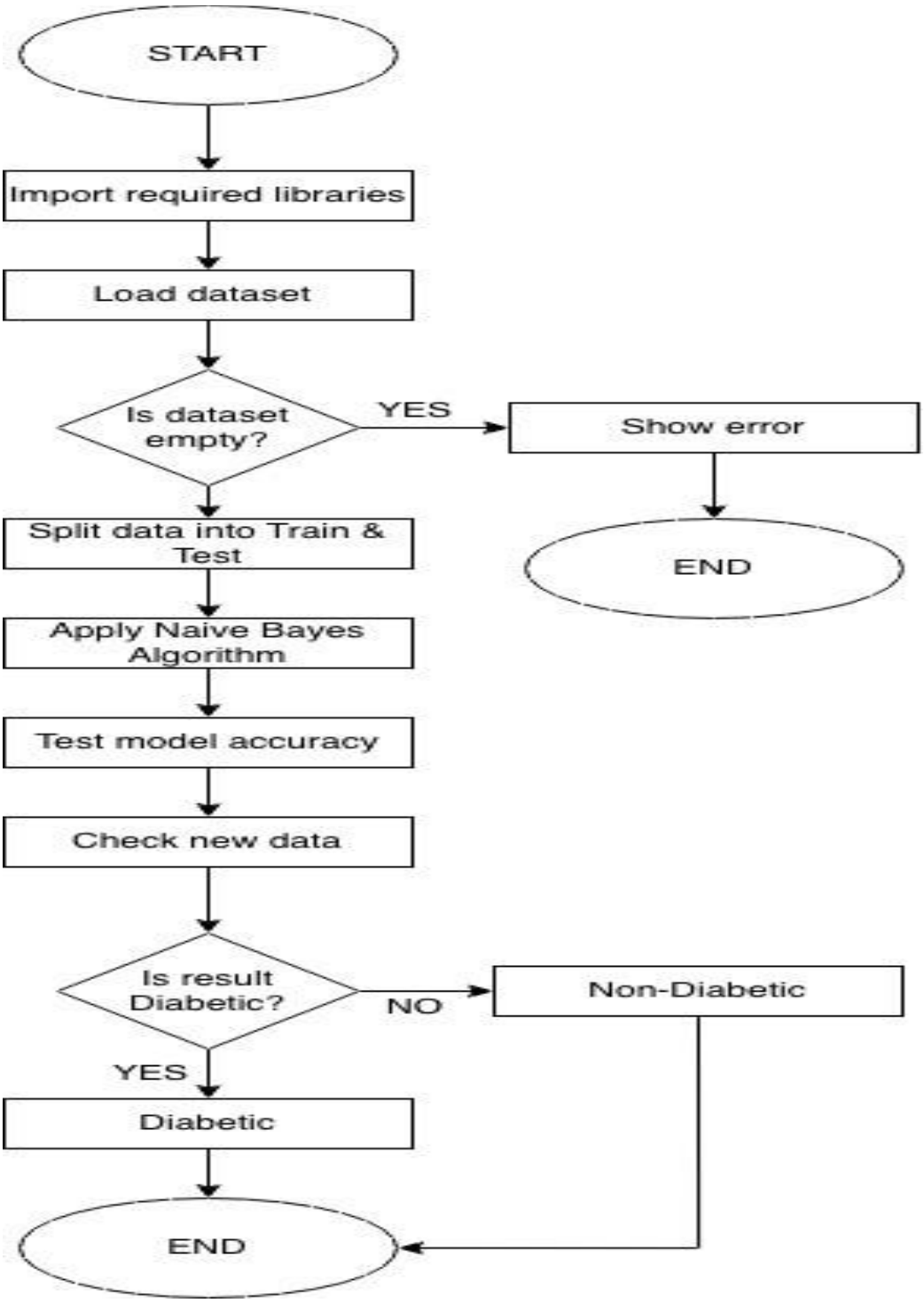

Figure 9 Naive Bayes Flowchart

4.4 State Transition Diagram

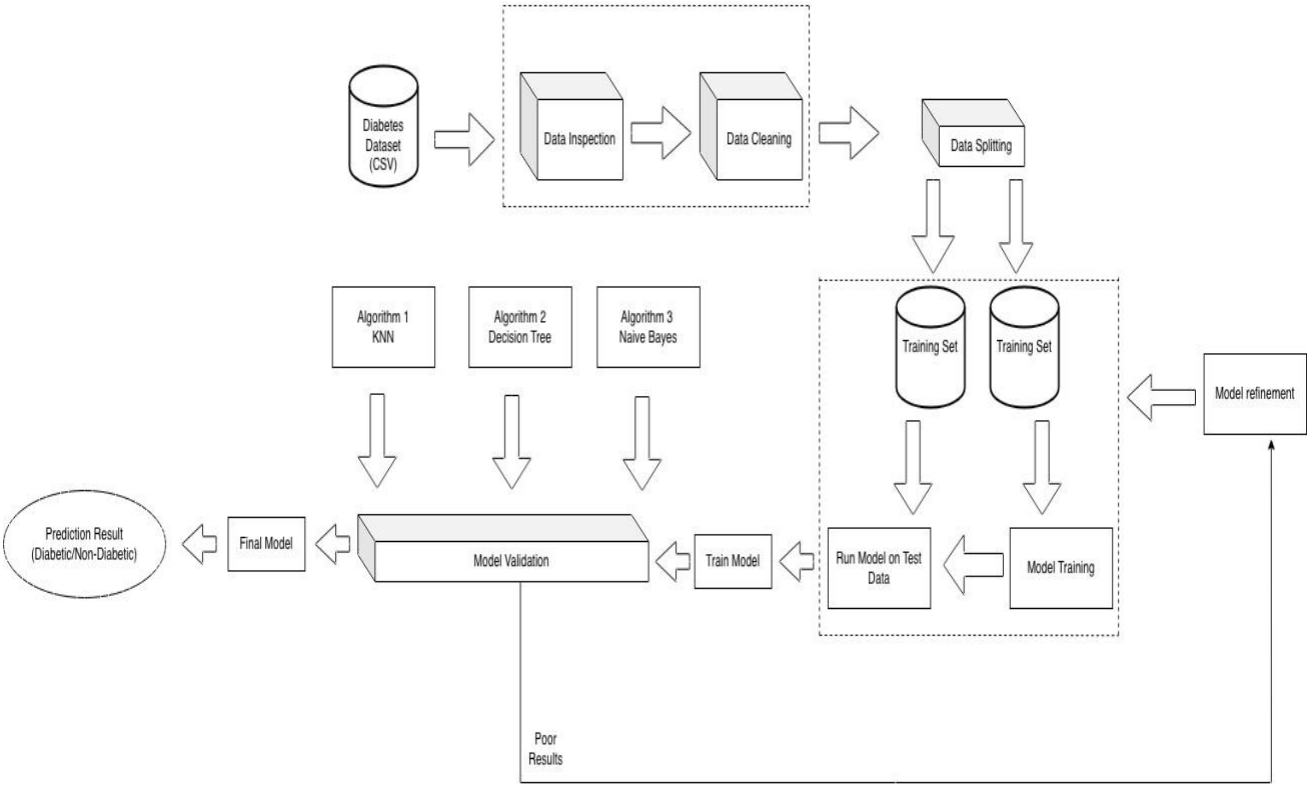

Figure 10 State Transition Diagram

5. Explanation of the Development Process

The development of this assignment followed a standard machine learning pipeline, incorporated step by step to make sure about the clarity and correctness. The process started with loading and inspecting the dataset to know its structure and attributes. Data preprocessing was then performed, which includes checking missing values and converting categorical values into numerical form to make the data refined for machine learning models.

After preprocessing, the dataset was parted into different input features and a target variable representing diabetes status. The data was divided into training and testing sets to let the fair evaluation of the models. Feature scaling was applied to standardise the input values, which is particularly crucial for distance-based algorithms. At last, different supervised learning models were trained, tested and evaluated by using accuracy as the performance metric.

5.1 Explanation of Tools & Technologies Used

There are two different tools and technologies: -

5.1.1. Python

Python was utilized as the most crucial programming language to implement data preprocessing machine learning models, and result evaluation for this assignment due to its simplicity and effectiveness in machine learning tasks.

5.1.2. Jupyter Notebook

Jupyter Notebook was utilized as the development environment to maintain the workflow, execute the code step by step, and show the results in a clear and readable way.

5.2 Libraries Used

5.2.1 Pandas

The panda's library is utilized for uploading the dataset, analyzing its structure, and performing data manipulation tasks such as handling categorical variables, selecting columns and preparing the data for machine learning models.

5.2.2 Numpy

Numpy was utilized to support the numerical operations and array-based computations during data processing and model evaluation, giving effective handling of numerical data.

5.2.3 Scikit-learn (Sklenar)

Scikit-learn library was utilized as the primary machine learning framework in this assignment. It provided tools for data preprocessing, model training and evaluation.

1.Train_test_split

Helps to divide the dataset into training and testing sets for unbiased model evaluation.

2. Standard Scaler

Utilized to standardize feature values to a common scale, improving model performance.

3.Label Encoder

Used to convert categorical target labels into the numerical format.

4.KneighborsClassifier

Utilized for the K-Nearest Neighbours Classifiers classification model.

5.GaussianNB

Utilized to implement the Naïve Bayes classification model.

6.DecisionTreeClassifier

Utilized to implement the Decision tree classification model.

7.Accuracy score

Utilized to evaluate and compare model performance on prediction accuracy.

6. Achieved Results

The created machine learning application was successfully implemented and executed using Jupyter Notebook. First the dataset was uploaded, pre-processed, and prepared for model training through categorical encoding, feature scaling, and data splitting.

Three supervised machine learning models were split, trained and tested on the dataset K-Nearest Neighbours, Naïve Bayes, Decision Tree. The application gave accuracy results for all three models allowing performance comparison. Although, the Decision Tree got the highest accuracy, followed by KNN while Naïve Bayes got comparatively lower accuracy.

Screenshots of the application execution and the results are provided below:-

6.1 Environment Setup & Library Import

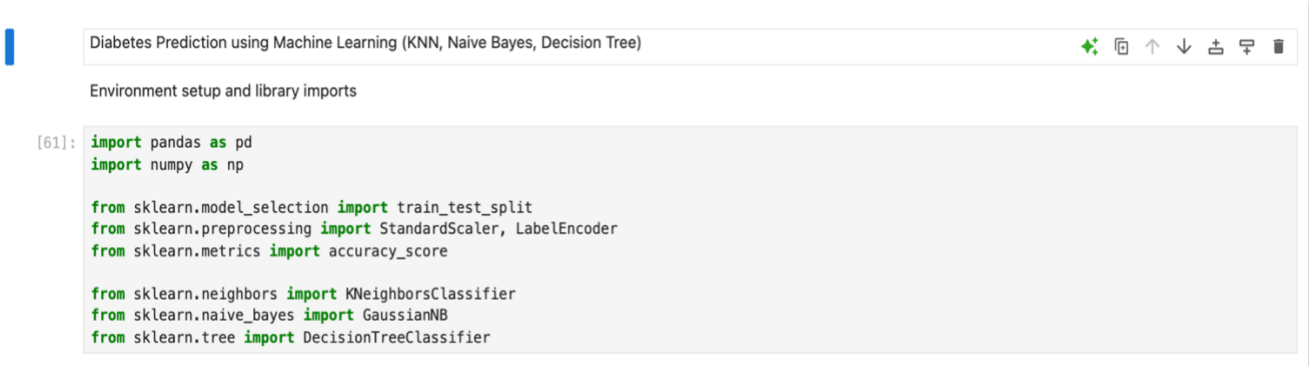

Figure 11 Environment setup

- The Required libraries are imported for data processing and machine learning.

6.2 Import CSV File

The dataset is loaded from a CSV file, and the first few rows are shown to understand its structure.

```
[70]: df = pd.read_csv("diabetes_health_indicators.csv")
      df.head()
      #Imported the dataset
```

[70]:

	ID	BMI	PhysHlth	Age	HighBP	HighChol	CholCheck	Smoker	Stroke	HeartDiseaseorAttack	...	HvyAlcoholConsump	AnyHealthcare	NoDocbcCost
0	114414	29	0	65 to 69	0	1	1	0	0	0	...	0	1	1
1	168896	32	0	80 or older	1	1	1	0	0	0	...	0	1	0
2	68354	25	5	65 to 69	1	0	1	1	0	0	...	0	1	0
3	121194	24	0	80 or older	1	0	1	0	0	0	...	0	1	0
4	141150	31	0	25 to 29	0	0	1	0	0	1	...	0	1	0

5 rows × 23 columns

Figure 12 CSV Import

- The dataset is loaded from a CSV file and the first few rows are displayed.

6.3 Dataset loaded for Rows X Column Confirmation

The full dataset is displayed to confirm that all rows and columns were loaded correctly.

```
[72]: df
      #Dataset displayed
```

[72]:

	ID	BMI	PhysHlth	Age	HighBP	HighChol	CholCheck	Smoker	Stroke	HeartDiseaseorAttack	...	HvyAlcoholConsump	AnyHealthcare	NoDocbc
0	114414	29	0	65 to 69	0	1	1	0	0	0	...	0	1	
1	168896	32	0	80 or older	1	1	1	0	0	0	...	0	1	
2	68354	25	5	65 to 69	1	0	1	1	0	0	...	0	1	
3	121194	24	0	80 or older	1	0	1	0	0	0	...	0	1	
4	141150	31	0	25 to 29	0	0	1	0	0	1	...	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
194820	43401	28	28	50 to 54	1	1	1	0	0	0	...	0	1	
194821	233048	25	0	60 to 64	0	0	1	1	0	0	...	0	1	
194822	1241	33	1	65 to 69	0	1	1	1	0	1	...	0	1	
194823	248165	33	0	60 to 64	0	0	1	0	0	0	...	0	1	
194824	133109	30	2	60 to 64	0	1	1	0	0	0	...	0	1	

194825 rows × 23 columns

Figure 13 CSV Load confirmation

-The full dataset is shown to confirm that all rows and columns were loaded correctly.

6.4 Checking the Datatypes

This step checks the data type of each column to identify which columns contain categorical values that need to be converted into numerical form.

```
[78]: df.dtypes
      #Checking the data types
```

```
[78]: ID                       int64
      BMI                      int64
      PhysHlth                 int64
      Age                     object
      HighBP                   int64
      HighChol                 int64
      CholCheck                int64
      Smoker                   int64
      Stroke                   int64
      HeartDiseaseorAttack     int64
      PhysActivity             int64
      Fruits                   int64
      Veggies                  int64
      HvyAlcoholConsump        int64
      AnyHealthcare            int64
      NoDocbcCost              int64
      GenHlth                 object
      MentHlth                 int64
      DiffWalk                 int64
      Sex                      int64
      Education               object
      Income                   int64
      Diabetes_binary         object
      dtype: object
```

Figure 14 Check datatypes

-This step checks the datatype of each column and identify which column has the categorical values.

6.5 Extracted Categorical Columns

Checking Categorical Columns

```
[13]: df[["Age", "GenHlth", "Education", "Diabetes_binary"]].head()
      #Extracted categorical values column
```

[13]:

	Age	GenHlth	Education	Diabetes_binary
0	65 to 69	Poor	6	Non-Diabetic
1	80 or older	Poor	Some College Degree	Non-Diabetic
2	65 to 69	Fair	6	Non-Diabetic
3	80 or older	Very Good	Advanced Degree	Non-Diabetic
4	25 to 29	Very Good	6	Diabetic

Figure 15 Extracted categorical values

-The extracted categorical columns are being checked.

6.6 Checking the Columns before Encoding

This step checks the data types of the selected columns to confirm that they are categorical before encoding.

```
[15]: df[["Age", "GenHlth", "Education", "Diabetes_binary"]].dtypes
      #Categorical columns datatypes
```

```
[15]: Age               object
      GenHlth           object
      Education         object
      Diabetes_binary   object
      dtype: object
```

Figure 16 Checking the categorical columns

-This step checks the data types of the selected columns to finalize they arw categorical before the encoding starts.

6.7 Encoding Output Label Separately

The Diabetes Binary column is encoded separately because it shows output label, while the other categorical features are processed in the next step.

```
[17]: le = LabelEncoder()
      df["Diabetes_binary_num"] = le.fit_transform(df["Diabetes_binary"])

      df[["Diabetes_binary", "Diabetes_binary_num"]].head()
      #Shows the output label
```

[17]:

	Diabetes_binary	Diabetes_binary_num
0	Non-Diabetic	1
1	Non-Diabetic	1
2	Non-Diabetic	1
3	Non-Diabetic	1
4	Diabetic	0

Figure 17 Encoding output label seperate

-The Diabetes binary columns is encoded separately because it shows the output label whereas others are input.

6.8 One-Hot Encoding the Input Labels

The remaining categorical input features are converted into numerical form using one-hot encoding so they can be used by the machine learning models.

```
[68]: X = df.drop(columns=["ID", "Diabetes_binary", "Diabetes_binary_num"])
      y = df["Diabetes_binary_num"]
      X = pd.get_dummies(X, columns=["Age", "GenHlth", "Education"], drop_first=True)

      X.head()
      #Used one-hot encoding
```

[68]:

	BMI	PhysHlth	HighBP	HighChol	CholCheck	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	...	Age_80 or older	GenHlth_Fair	GenHlth_Good	GenHlth
0	29	0	0	1	1	0	0	0	1	0	...	False	False	False	
1	32	0	1	1	1	0	0	0	0	1	...	True	False	False	
2	25	5	1	0	1	1	0	0	1	1	...	False	True	False	
3	24	0	1	0	1	0	0	0	0	0	...	True	False	False	
4	31	0	0	0	1	0	0	1	1	1	...	False	False	False	

5 rows × 39 columns

Figure 18 One-Hot encoding

-The remaining input labels which has categorical values is being one-hot encoded to be used by machine learning models.

6.9 Splitting the Datasets & Feature Scaling

Splitting the Dataset into Training and Testing Sets

```
[28]: X_train, X_test, y_train, y_test = train_test_split(
          X, y,
          test_size=0.2,
          random_state=42,
          stratify=y
      )

      X_train.shape, X_test.shape
      #Splitted the datasets into training and testing sets
```

```
[28]: ((155860, 39), (38965, 39))
```

Feature Scaling

```
[32]: scaler = StandardScaler()

      X_train_scaled = scaler.fit_transform(X_train)
      X_test_scaled = scaler.transform(X_test)
      #Feature scaling to bring all values to a similar range for fair model training
```

Figure 19 Split dataset & Feature Scale

-In this step the dataset was splitted into training and testing sets and then the feature scaling was also done in order to bring all values to a similar range for fair model training.

6.10 KNN Model Trained & Tested

KNN Model Training and Testing

```
[35]: knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train_scaled, y_train)

knn_pred = knn.predict(X_test_scaled)
knn_accuracy = accuracy_score(y_test, knn_pred)

knn_accuracy
#KNN model training and testing done
```

```
[35]: 0.8463236237649172
```

Figure 20 KNN Trained & Tested

-One of the algorithms KNN Model was trained and tested in this step

6.11 Navies Bayes Model Trained & Tested

Naive Bayes Model Training and Testing

```
[38]: nb = GaussianNB()
nb.fit(X_train_scaled, y_train)

nb_pred = nb.predict(X_test_scaled)
nb_accuracy = accuracy_score(y_test, nb_pred)

nb_accuracy
#Naive Bayes training and testing done
```

```
[38]: 0.708584627229565
```

Figure 21 Naive Bayes Trained & Tested

- One of the algorithms Naïve Bayes was trained and tested in this step.

6.12 Decision Tree Model Trained & Tested

Decision Tree Model Training and Testing

```
[41]: dt = DecisionTreeClassifier(random_state=42, max_depth=6)
      dt.fit(X_train_scaled, y_train)

      dt_pred = dt.predict(X_test_scaled)
      dt_accuracy = accuracy_score(y_test, dt_pred)

      dt_accuracy
      #Decision Tree training and testing done
```

```
[41]: 0.8628256127293725
```

Figure 22 Decision tree Trained & Tested

-One of the algorithms Decision Tree was trained and tested in this step.

6.13 Model Training Confirmation/Prediction/Test Data Size

Model Training Confirmation

```
[46]: print("KNN model:", knn)
      print("Naive Bayes model:", nb)
      print("Decision Tree model:", dt)
      #Trained the model and got the confirmation
```

```
KNN model: KNeighborsClassifier()
Naive Bayes model: GaussianNB()
Decision Tree model: DecisionTreeClassifier(max_depth=6, random_state=42)
```

Model Testing Confirmation(Predictions)

```
[50]: print("Sample predictions (first 10 values):\n")

      print("KNN:", knn_pred[:10])
      print("Naive Bayes:", nb_pred[:10])
      print("Decision Tree:", dt_pred[:10])
      #Confirmation done and Predictions too
```

```
Sample predictions (first 10 values):

KNN: [1 1 1 1 1 1 1 1 1 1]
Naive Bayes: [1 1 1 0 1 1 1 1 1 1]
Decision Tree: [1 1 1 1 1 1 1 1 1 1]
```

Test Data Size

```
[53]: print("Test data size check:\n")

      print("y_test:", len(y_test))
      print("KNN predictions:", len(knn_pred))
      print("Naive Bayes predictions:", len(nb_pred))
      print("Decision Tree predictions:", len(dt_pred))
      #Tested the data size
```

```
Test data size check:

y_test: 38965
KNN predictions: 38965
Naive Bayes predictions: 38965
Decision Tree predictions: 38965
```

Figure 23 Training confirmation/Prediction/ Test data size

-This step finalizes that the KNN Model, Naïve Bayes and Decision Tree were successfully trained. Sample prediction and the test data size are shown to verify that all models generated outputs correctly based on the test dataset.

6.14 Model Accuracy Evaluation & Comparison

Model Accuracy Evaluation and Comparison

```
[56]: accuracy_results = pd.DataFrame({
          "Model": ["KNN", "Naive Bayes", "Decision Tree"],
          "Accuracy": [knn_accuracy, nb_accuracy, dt_accuracy]
      })

      accuracy_results
      #Accuracy Evaluated and Comparison done
```

[56]:

	Model	Accuracy
0	KNN	0.846324
1	Naive Bayes	0.708585
2	Decision Tree	0.862826

Figure 24 Accuracy Test

-This step compares the accuracy of all three model to know the best performing algorithm.

6.15 Precision Evaluation & Comparison

Precision Calculation

```
In [37]: # Calculated precision for each model
         knn_precision = precision_score(y_test, knn_pred)
         nb_precision = precision_score(y_test, nb_pred)
         dt_precision = precision_score(y_test, dt_pred)

         print("Precision Scores:")
         print("KNN:", knn_precision)
         print("Naive Bayes:", nb_precision)
         print("Decision Tree:", dt_precision)
         # Precision model was calculated for each model
```

```
Precision Scores:
KNN: 0.8791148299020147
Naive Bayes: 0.9400817427879846
Decision Tree: 0.8765930161105026
```

Figure 25 Precision Test

-This precision calculation shows the precision score of all 3-classification model Decision Tree, Naïve Bayes & KNN.

6.16 Recall Sensitivity

Recall (Sensitivity)

```
In [39]: # Recall (Sensitivity) calculation for each model
knn_recall = recall_score(y_test, knn_pred, zero_division=0)
nb_recall  = recall_score(y_test, nb_pred,  zero_division=0)
dt_recall  = recall_score(y_test, dt_pred,  zero_division=0)

print("Recall (Sensitivity) Scores:")
print("KNN:", knn_recall)
print("Naive Bayes:", nb_recall)
print("Decision Tree:", dt_recall)
```

```
Recall (Sensitivity) Scores:
KNN: 0.9524093511450382
Naive Bayes: 0.7064348759541985
Decision Tree: 0.9783516221374046
```

Figure 26 Recall

-This recall(sensitivity) measures the recall of each model to calculate how efficiently they identify positive cases.

6.17 F1-Score

F1-Score Calculation

```
In [41]: # F1-Score calculation for each model
knn_f1 = f1_score(y_test, knn_pred, zero_division=0)
nb_f1  = f1_score(y_test, nb_pred,  zero_division=0)
dt_f1  = f1_score(y_test, dt_pred,  zero_division=0)

print("F1-Score Results:")
print("KNN:", knn_f1)
print("Naive Bayes:", nb_f1)
print("Decision Tree:", dt_f1)
```

```
F1-Score Results:
KNN: 0.9142955287112842
Naive Bayes: 0.806680627202615
Decision Tree: 0.9246811808638061
```

Figure 27 F1-Score

-This step measures the F1-Score of each models and gives a balanced measure of precision and the recall.

6.18 ROC-AUC

ROC-AUC

```
[43]: knn_auc = roc_auc_score(y_test, knn.predict_proba(X_test)[:, 1])
nb_auc  = roc_auc_score(y_test, nb.predict_proba(X_test)[:, 1])
dt_auc  = roc_auc_score(y_test, dt.predict_proba(X_test)[:, 1])

print("ROC-AUC Scores")
print("KNN:", knn_auc)
print("Naive Bayes:", nb_auc)
print("Decision Tree:", dt_auc)
#ROC - AUC scores comparison done
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/base.py:486: UserWarning: X
t feature names
  warnings.warn(
ROC-AUC Scores
KNN: 0.5978557837495202
Naive Bayes: 0.7021231459786572
Decision Tree: 0.7760606477801923
```

Figure 28 ROC – AUC

-This measures the ROC-AUC scores of each models to calculate how well they differentiate between positive and negative classes, the warning occurs because the test data was passed without feature names.

6.19 Log Loss (Cross Entropy Loss)

Log Loss

```
74]: knn_logloss = log_loss(y_test, knn.predict_proba(X_test))
     nb_logloss  = log_loss(y_test, nb.predict_proba(X_test))
     dt_logloss  = log_loss(y_test, dt.predict_proba(X_test))

     print("Log Loss (Cross-Entropy Loss)")
     print("KNN:", knn_logloss)
     print("Naive Bayes:", nb_logloss)
     print("Decision Tree:", dt_logloss)
     # Log Loss comparison done
```

```
/opt/anaconda3/lib/python3.12/site-packages/sklearn/base.py:486: UserWarning: X
t feature names
  warnings.warn(
Log Loss (Cross-Entropy Loss)
KNN: 3.1217806770541916
Naive Bayes: 31.020397110205742
Decision Tree: 0.3527433490969214
```

Figure 29 Log Loss

-This measure the log loss of each model to calculate the error in predicted probabilities, and the warning occurs as the test data was used without feature names.

6.20 Confusion Matrix

Confusion Matrix

```
[47]: print("Confusion Matrix - KNN")
      print(confusion_matrix(y_test, knn_pred))

      print("\nConfusion Matrix - Naive Bayes")
      print(confusion_matrix(y_test, nb_pred))

      print("\nConfusion Matrix - Decision Tree")
      print(confusion_matrix(y_test, dt_pred))
      #Shows the correct and incorrect predictions
```

```
Confusion Matrix - KNN
[[ 1037  4392]
 [ 1596 31940]]

Confusion Matrix - Naive Bayes
[[ 3919  1510]
 [ 9845 23691]]

Confusion Matrix - Decision Tree
[[  810  4619]
 [  726 32810]]
```

Figure 30 Confusion Matrix

-This step shows the confusion matrix of each model to display the numbers of correct and incorrect predictions among the predicted classes.

7. Application

The developed application is given as a runnable Python-based solution using the Jupyter Notebook. All the source code files (.ipynb & .py) and also the dataset is organized into a single zip folder to make sure of the error free execution. The application runs successfully executed, and screenshots of every processing step and output results are included in this particular report as evidence of the correct functionality.

8. Conclusion

The project successfully applied the machine learning concepts to create and measure classification models using a structured workflow. The diabetes dataset was prepared and preprocessed and split into the training and the testing sets, after which three classification algorithms (KNN, Naïve Bayes, Decision Tree) were done. Model performance was measured using the multiple metrics including accuracy, precision, recall, F1-Score, ROC-AUC, Log loss and lastly the confusion matrix analysis, which allowed the comparison of results rather than depending on a single measure.

The measurement results shows that the Decision tree performed most consistent among most metrics, showing strong predictive capability and effective handling of class separation. KNN also displayed competitive performance, while Naïve Bayes produced lower results, due to its simplifying assumptions about feature independence. Overall, this project shows the importance of using the multiple evaluation metrics when assessing machine learning models and shows the solid understanding of model development, evaluation and interpretation. The project provides a foundation for further improvements such as model optimization, feature improvement or the application of more advanced algorithms.

9.Reference

Works Cited

Centers for Disease Control and Prevention, 2023. *Diabetes Risk Factors.* [Online] Available at: https://www.cdc.gov/diabetes/basics/risk-factors.html [Accessed 17 dec 2025].

International Diabetes Federation, 2023. *IDF Diabetes Atlas.* [Online] Available at: https://diabetesatlas.org [Accessed 17 dec 2025].

International Diabetes Federation, 2023. *IDF Diabetes Atlas.* [Online] Available at: https://diabetesatlas.org [Accessed 16 dec 2025].

Nepal Health Research Council, 2023. *Non-Communicable Diseases Risk Factors: STEPS Survey Nepal.* [Online] Available at: https://nhrc.gov.np [Accessed 17 DEC 2025].

World Health Organization, 2023. *Diabetes.* [Online] Available at: https://www.who.int/news-room/fact-sheets/detail/diabetes [Accessed 17 dec 2025].

World Health Organization, 2023. *Diabetes.* [Online] Available at: https://www.who.int/news-room/fact-sheets/detail/diabetes [Accessed 17 dec 2025].

World Health Organization, 2023. *STEPwise Approach to Noncommunicable Disease Risk Factor Surveillance (STEPS) – Nepal.* [Online] Available at: https://www.who.int/teams/noncommunicable-diseases/surveillance/systemstools/steps [Accessed 17 dec 2025].