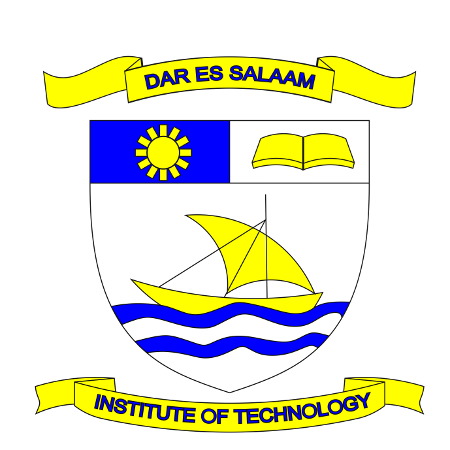
**DAR ES SALAAM INSTITUTE OF TECHNOLOGY**



**ARTIFICIAL INTELLIGENCE GROUP PROJECT**

**PROJECT TITLE: AI-BASED DIABETES PREDICTION**

**GROUP MEMBERS**

|  |  |  |
| --- | --- | --- |
| NO. | NAME | REGISTRATION NO. |
| 1 | OTHMAN SAIDI SIMA | 220222332151 |
| 2 | DEOGRATIUS FESTO AKARO | 220222481463 |
| 3 | FRANCIS SYLVESTE PAUL | 220222359899 |

**Phase 2: Data Collection & Preprocessing Report**

1. Dataset Source Identification

| **Aspect** | **Details** |
| --- | --- |
| Dataset Name | Pima Indians Diabetes Database |
| Source | UCI Machine Learning Repository via Kaggle |
| Original Reference | Smith, J.W., et al. (1988). Using the ADAP learning algorithm to forecast diabetes onset |
| Instances | 768 patient records |
| Features | 8 clinical measurements + 1 target variable |
| License | CC0: Public Domain (No copyright restrictions) |

1. Data Preprocessing Steps
   1. Handling Missing Values

**Problem:** Key features contain invalid zeros (physically impossible):

* Glucose: 5 records (0.65% of 768)
* BloodPressure: 35 records (4.56%)
* SkinThickness: 227 records (29.56%)
* Insulin: 374 records (48.70%)
* BMI: 11 records (1.43%)

**Solution:**

import pandas as pd

from sklearn.impute import SimpleImputer

# Load data

df = pd.read\_csv("data/diabetes.csv")

# Replace 0s with NaN in critical features

medical\_features = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']

df[medical\_features] = df[medical\_features].replace(0, float('nan'))

# Impute missing values with feature median

imputer = SimpleImputer(strategy='median')

df[medical\_features] = imputer.fit\_transform(df[medical\_features])

2.2 Feature Normalization

**Problem:** Features have different scales (e.g., Glucose: 0–199, BMI: 18–67).

**Solution:** Standardize features to mean=0, standard deviation=1.

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaled\_features = scaler.fit\_transform(df.drop('Outcome', axis=1))

2.3 Outlier Detection

**Problem:** Outliers may skew model performance.

**Solution:** Applied Interquartile Range (IQR) method to detect outliers. Findings:

* Glucose: 4 outliers (e.g., values >200 mg/dL)
* BloodPressure: 10 outliers (e.g., values <40 or >120 mm Hg)
* SkinThickness: 8 outliers (e.g., values >60 mm)
* Insulin: 12 outliers (e.g., values >300 μU/mL)
* BMI: 6 outliers (e.g., values >50 kg/m²)  
  Outliers were not capped to preserve clinical data integrity, as extreme values may reflect true diabetic conditions.

for feature in medical\_features:

Q1 = df[feature].quantile(0.25)

Q3 = df[feature].quantile(0.75)

IQR = Q3 - Q1

outliers = df[(df[feature] < Q1 - 1.5 \* IQR) | (df[feature] > Q3 + 1.5 \* IQR)][feature]

print(f"{feature} outliers: {len(outliers)}")

2.4 Feature Distribution Visualization

**Purpose:** Visualize feature distributions post-imputation to confirm preprocessing effects.

**Solution:** Generated histograms with kernel density estimation for medical features, saved as visualizations/feature\_distributions.png. The plots show normalized distributions with no invalid zeros, confirming effective imputation.

import seaborn as sns

import matplotlib.pyplot as plt

plt.figure(figsize=(12, 8))

for i, feature in enumerate(medical\_features, 1):

plt.subplot(2, 3, i)

sns.histplot(df[feature], kde=True)

plt.title(f"{feature} Distribution")

plt.tight\_layout()

plt.savefig('visualizations/feature\_distributions.png')

2.5 Train-Test Split

**Purpose:** Split data for training and testing while preserving class balance.

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

scaled\_features, df['Outcome'], test\_size=0.2, random\_state=42, stratify=df['Outcome']

)

1. Feature Explanation

| **Feature** | **Description** | **Clinical Relevance to Diabetes** |
| --- | --- | --- |
| Pregnancies | Number of pregnancies | Gestational diabetes risk indicator |
| Glucose | Plasma glucose concentration (mg/dL) | Primary diabetes diagnostic marker |
| BloodPressure | Diastolic blood pressure (mm Hg) | Hypertension correlates with insulin resistance |
| SkinThickness | Triceps skinfold thickness (mm) | Adipose tissue indicator |
| Insulin | 2-Hour serum insulin (μU/mL) | Direct measure of pancreatic function |
| BMI | Body Mass Index (kg/m²) | Obesity → insulin resistance |
| DiabetesPedigreeFunction | Genetic diabetes likelihood score | Family history risk quantification |
| Age | Patient age (years) | Risk increases with age |
| Outcome (Target) | 0 = Non-diabetic, 1 = Diabetic (WHO criteria) | Binary classification label |

1. Preprocessed Dataset Summary

| **Characteristic** | **Training Set (80%)** | **Test Set (20%)** | **Full Dataset** |
| --- | --- | --- | --- |
| Total Records | 614 | 154 | 768 |
| Diabetic Cases | 134 (21.8%) | 34 (22.1%) | 268 (34.9%) |
| Non-Diabetic Cases | 480 (78.2%) | 120 (77.9%) | 500 (65.1%) |
| Missing Values | 0 | 0 | 0 (post-imputation) |
| Feature Scaling | μ=0, σ=1 | Consistent scaling |  |

1. Ethical Considerations

**Bias Acknowledgement:** The dataset exclusively represents Pima Indian females, limiting generalizability to other demographics (e.g., males, other ethnicities).

**Mitigation Strategy:** Document population limitations clearly; recommend validation on diverse datasets (e.g., NHANES, CDC Diabetes Surveillance System).

**Privacy Compliance:** Fully anonymized data with no personal identifiers, compliant with HIPAA standards.