Assessment Report

on

"Diabetes Prediction"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

ARCHY MITTAL (202401100400046)

ASHUTOSH KUMAR GUPTA (202401100400058)

AVESH (202401100400060)

AYUSH KUMAR (202401100400064)

ABHAY PRATAP SINGH (202401100400005)

SEC – 'A'

Under the supervision of

"Bikki Gupta Sir"

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

With the rise of digital lending platforms, automating credit risk assessments through data-driven approaches has become essential. This project focuses on predicting loan defaults using **supervised machine learning**. By analyzing borrower data like credit scores, income, and loan history, the goal is to develop a model that assists financial institutions in making informed loan decisions.

---

2. Problem Statement

The challenge is to predict whether a borrower will default on a loan using available credit and financial history. Such a classification system helps lenders identify high-risk applicants and reduce lending risk.

---

3. Objectives

- Preprocess the dataset for ML training.
- Train a Logistic Regression model for loan default classification.

- Evaluate performance using metrics like accuracy, precision, recall, and F1-score.
- Visualize classification performance using a confusion matrix heatmap.

---

4. Methodology

Data Collection:

- A CSV dataset is uploaded by the user.

Data Preprocessing:

- Handle missing values (mean/mode imputation).
- One-hot encode categorical data.
- Apply feature scaling with Standard Scaler.

Model Building:

- Split data into training and testing sets.
- Train a **Logistic Regression** classifier.

Evaluation:

- Measure accuracy, precision, recall, and F1-score.
- Visualize the **confusion matrix** using a heatmap.

- **Model Building**:
  - Splitting the dataset into training and testing sets.
  - Training a Logistic Regression classifier.

- **Model Evaluation***:*
  - Evaluating accuracy, precision, recall, and F1-score.
  - Generating a confusion matrix and visualizing it with a heatmap.

---

5. Data Preprocessing

Missing numerical values: filled with column-wise mean.

Categorical values: transformed using one-hot encoding.

Feature scaling: done using Standard Scaler.

Train-test split: 80% for training, 20% for testing.

---

6. Model Implementation

Logistic Regression is selected due to its efficiency in binary classification.

Trained on the preprocessed dataset.

Used to predict loan default status on the test set.

---

7. Evaluation Metrics

- **Accuracy**: Overall prediction correctness.

- **Precision**: Correctness of predicted defaults.
- **Recall**: Ability to identify actual defaults.
- **F1 Score**: Balance between precision and recall.
- **Confusion Matrix**: Visualized using Seaborn to show prediction errors.

---

8. Results and Analysis

- The model demonstrated reasonable performance.
- The confusion matrix helped assess false positives/negatives.
- Precision and recall revealed how effectively the model identified defaults.

---

9. Conclusion

Logistic Regression successfully predicted loan defaults with acceptable accuracy.

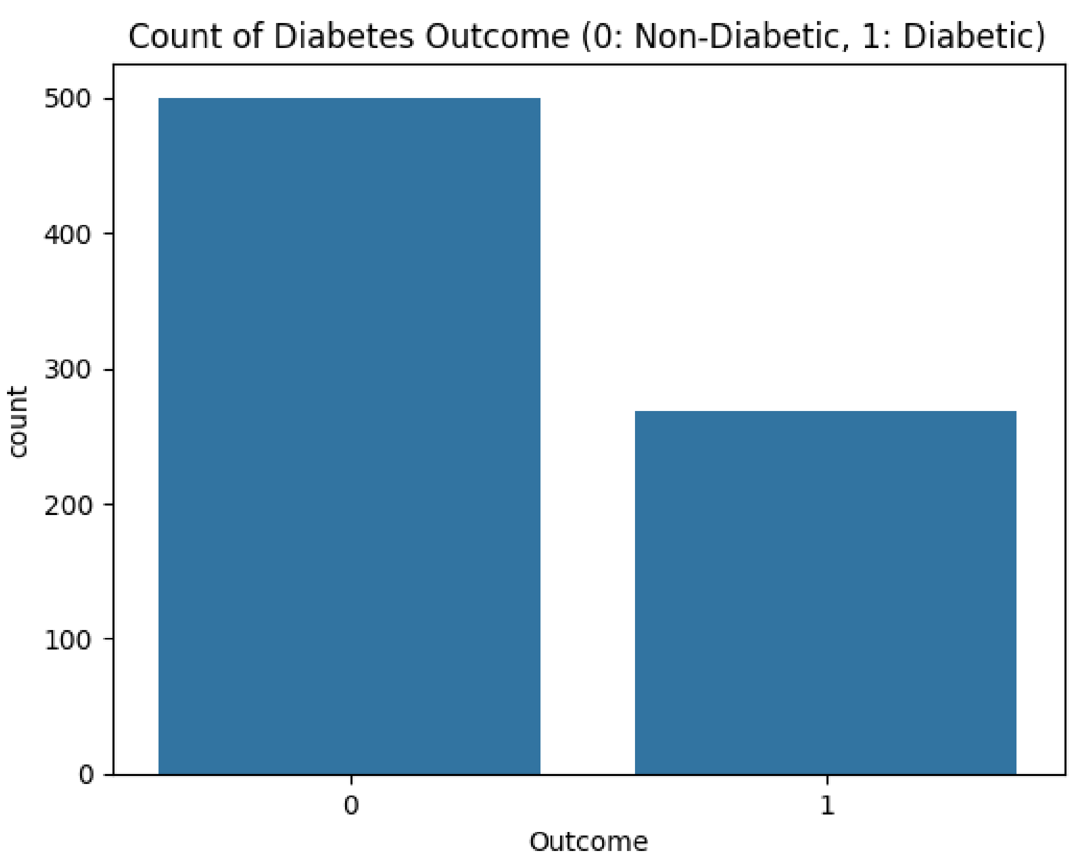
Demonstrated the potential of AI/ML in automating loan decisions and enhancing credit risk analysis.

Future improvements could involve more advanced algorithms and better handling of class imbalance.

#CODE

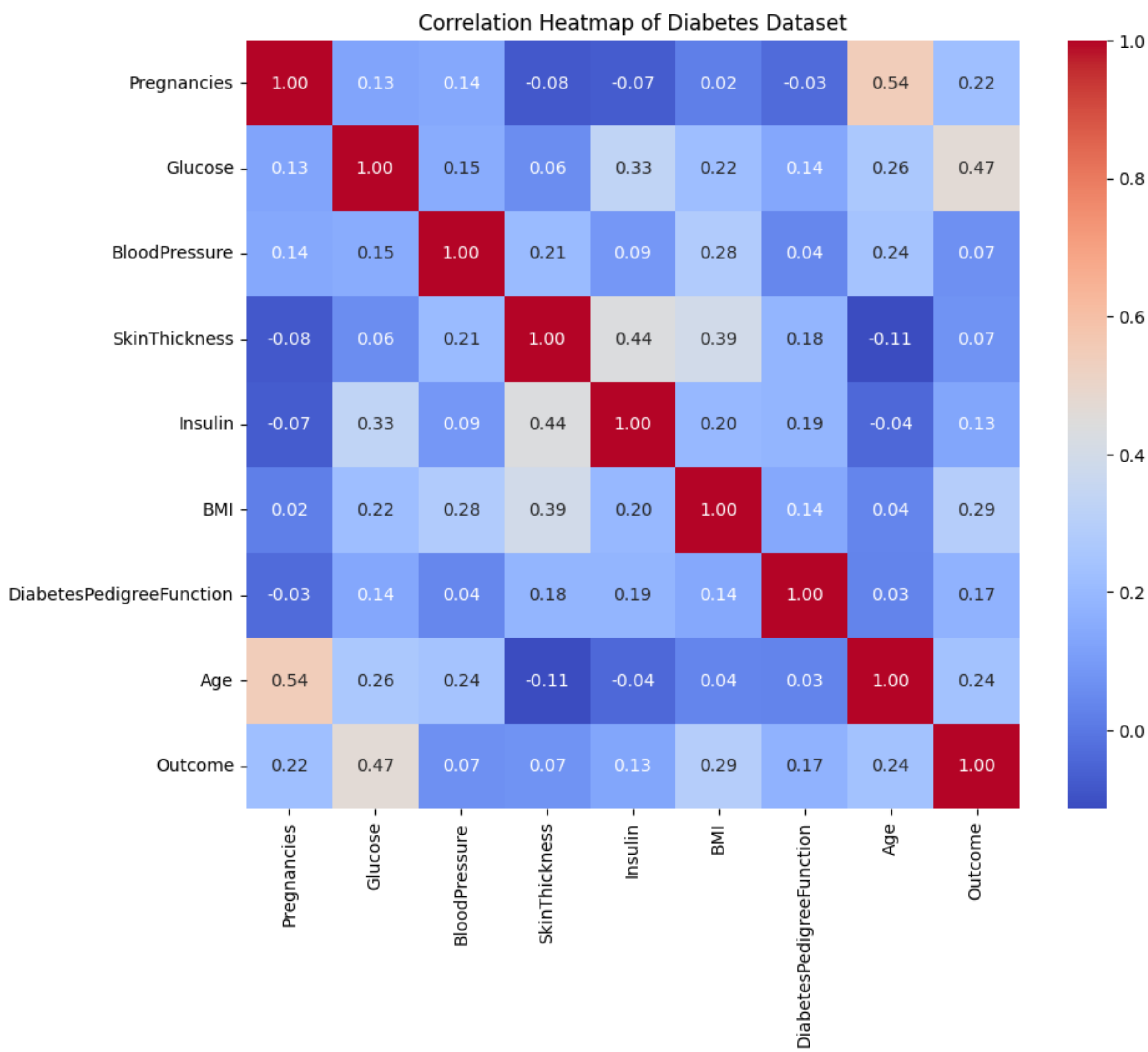
```
# Import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixDisplay


# Load dataset

df = pd.read_csv("/content/diabetes.csv")


 # Basic EDA

print("First 5 rows of dataset:")

df.head()

print("Last 5 rows of dataset:")

df.tail()


# Data Set Info

print("\nDataset Info:")

df.info()
```

```
# Summary Statistics
print("\nSummary Statistics:")
df.describe()

# Countplot of target variable
sns.countplot(x="Outcome", data=df)
plt.title("Count of Diabetes Outcome (0: Non-Diabetic, 1: Diabetic)")
plt.show()
```



Correlation heatmap

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

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

plt.title("Correlation Heatmap of Diabetes Dataset")

plt.show()
```



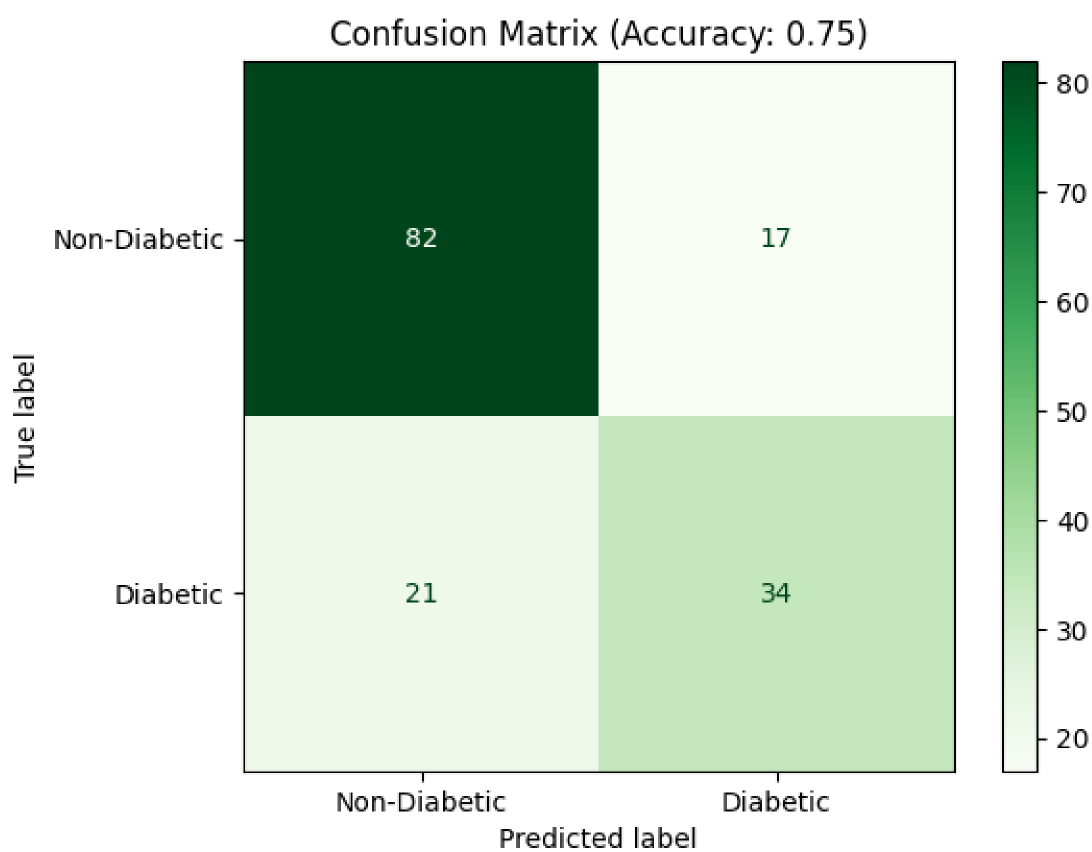
```
# Replace 0s with NaN in columns where 0 is not a valid value

cols_with_zero_invalid = ["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]

df[cols_with_zero_invalid] = df[cols_with_zero_invalid].replace(0, np.nan)

df.info()


# Fill missing values with median of each column

df.fillna(df.median(numeric_only=True), inplace=True)

df.info()


# Split into features and target

X = df.drop("Outcome", axis=1)

y = df["Outcome"]


# Train-test split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Standardize features

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)


# Model Training

model = LogisticRegression()

model.fit(X_train_scaled, y_train)
```

Evaluation

```
y_pred = model.predict(X_test_scaled)

accuracy = accuracy_score(y_test, y_pred)

print(f"\nModel Accuracy: {accuracy:.2f}")
```

Confusion Matrix Visualization

```
cm = confusion_matrix(y_test, y_pred)

disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Non-Diabetic", "Diabetic"])

disp.plot(cmap=plt.cm.Greens)

plt.title(f"Confusion Matrix (Accuracy: {accuracy:.2f})")

plt.show()
```



10. References

- scikit-learn documentation
- Pandas documentation
- Seaborn visualization library