Assessment Report

on

"Diabetes Prediction"

submitted as partial fulfilment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE (AI)

By:

Group Members:-

[Samarth Shukla] 202401100300212
[Sagar Shrivastava] 202401100300209
[Satwik Shaw] 202401100300217
[Prateek Kumar] 202401100300178
[Satyam Tiwari] 202401100300219

Section: [C]
Under the supervision of
[Mayank Lakhtoia]

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

As the prevalence of diabetes continues to rise, especially in developing nations, early prediction through data-driven healthcare is becoming increasingly vital. This project addresses the problem of predicting the likelihood of diabetes using supervised machine learning techniques. Utilizing a dataset of diagnostic attributes such as glucose level, BMI, age, and insulin levels, the model aims to support clinical decision-making through predictive analysis.

2. Problem Statement

To predict whether a patient is likely to be diagnosed with diabetes using available medical and personal health data. This classification task aims to assist healthcare providers in early diagnosis and preventive care.

3. Objectives

- Preprocess the dataset for training a machine learning model.
- Train a Logistic Regression model to classify diabetic and non-diabetic cases.
- Evaluate model performance using standard classification metrics.
- Visualize the confusion matrix using a heatmap for interpretability.

4. Methodology

Data Collection: The user uploads a CSV file containing the dataset (Pima Indians Diabetes Dataset).

Data Preprocessing:

- Handling missing values using mean imputation for columns with zero entries.
- Feature scaling using StandardScaler.

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

The dataset is cleaned and prepared as follows:

- Missing numerical values in fields such as Glucose and BloodPressure are replaced with column means.
- Data is scaled using StandardScaler to normalize feature values.
- The dataset is split into 80% training and 20% testing.

6. Model Implementation

Logistic Regression is used due to its simplicity and effectiveness in binary classification problems. The model is trained on the processed dataset and used to predict diabetes status on the test set.

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy**: Measures overall correctness.
- **Precision**: Indicates the proportion of predicted positives that are actual positives.
- **Recall**: Shows the proportion of actual positives that were correctly identified.
- **F1 Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: Visualized using Seaborn heatmap to understand prediction errors.

8. Results and Analysis

- The model provided reasonable performance on the test set.
- Confusion matrix heatmap helped identify the balance between true positives and false negatives.
- Precision and recall indicated how well the model detected diabetic patients versus false negatives.

9. Conclusion

The logistic regression model successfully classified diabetic and non-diabetic cases with satisfactory performance metrics. The project demonstrates the potential of using machine learning for automating healthcare risk assessments. However, improvements can be made by exploring more advanced models and handling data imbalance.

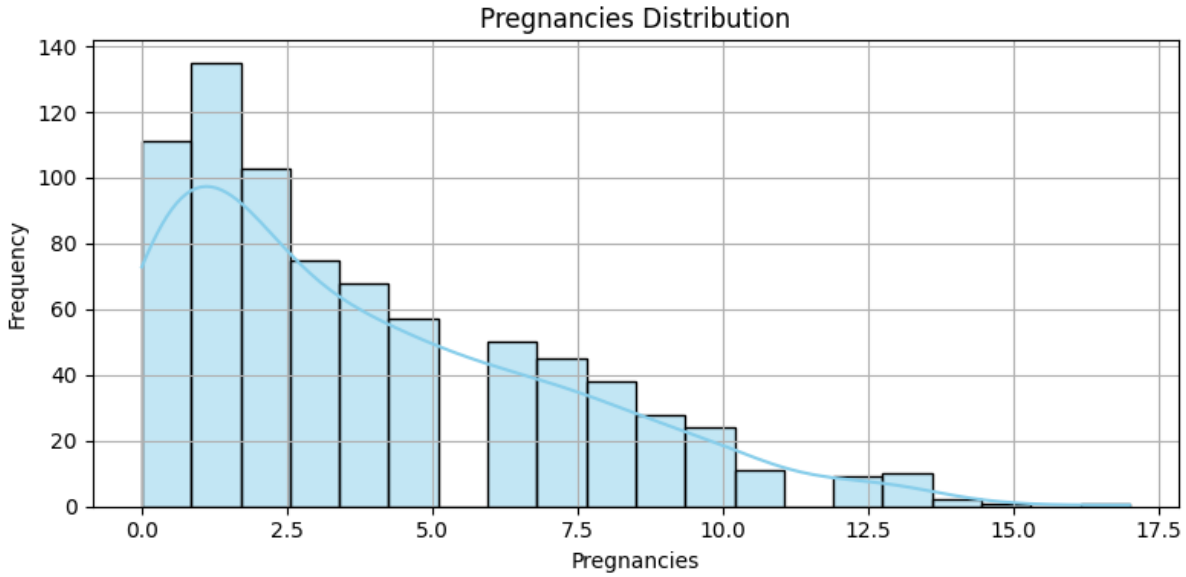
10. References

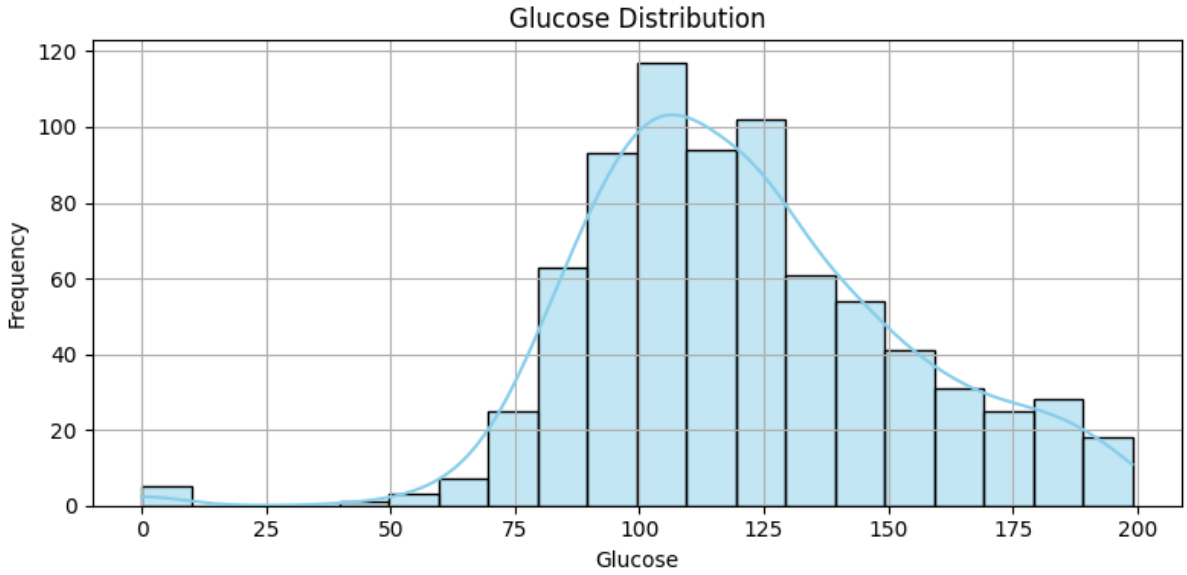
- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on diabetes prediction

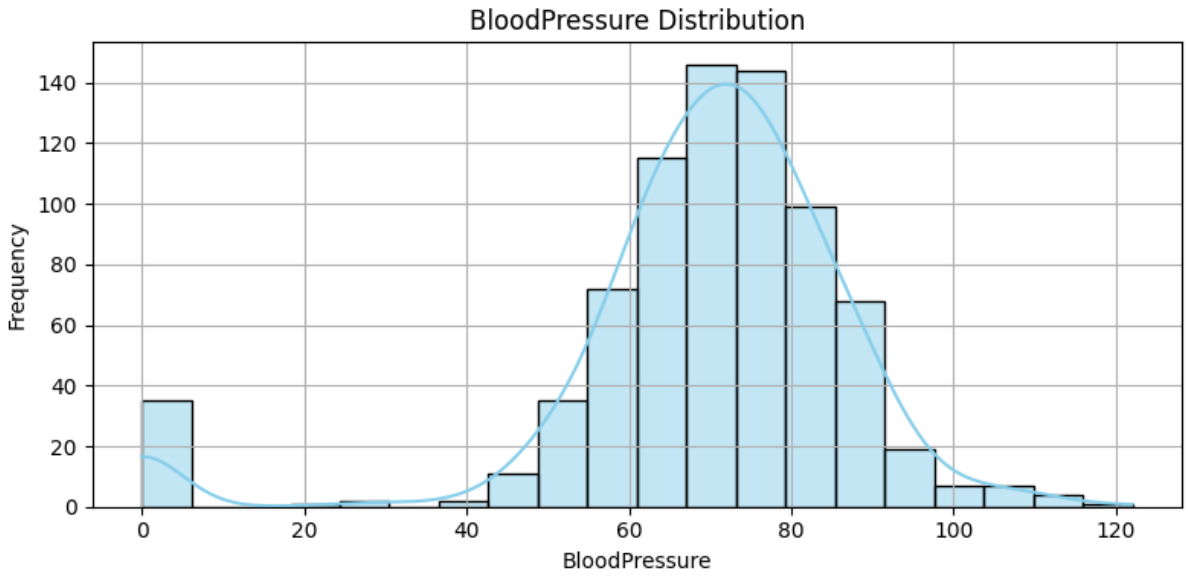
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixDisplay, classification_report
import time

# Load dataset
df = pd.read_csv('diabetes_new.csv')

# === EDA: Distribution Plots for each feature ===
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns.tolist()
numeric_columns.remove('Outcome')  # Exclude target column

for col in numeric_columns:
    plt.figure(figsize=(8, 4))
    sns.histplot(df[col], bins=20, kde=True, color='skyblue', edgecolor='black')
    plt.title(f'{col} Distribution')
    plt.xlabel(col)
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.tight_layout()
    plt.show()

# === EDA: Correlation Heatmap ===
plt.figure(figsize=(10, 8))
corr = df.corr()
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm', square=True)
plt.title("Feature Correlation Heatmap")
plt.tight_layout()
plt.show()
```

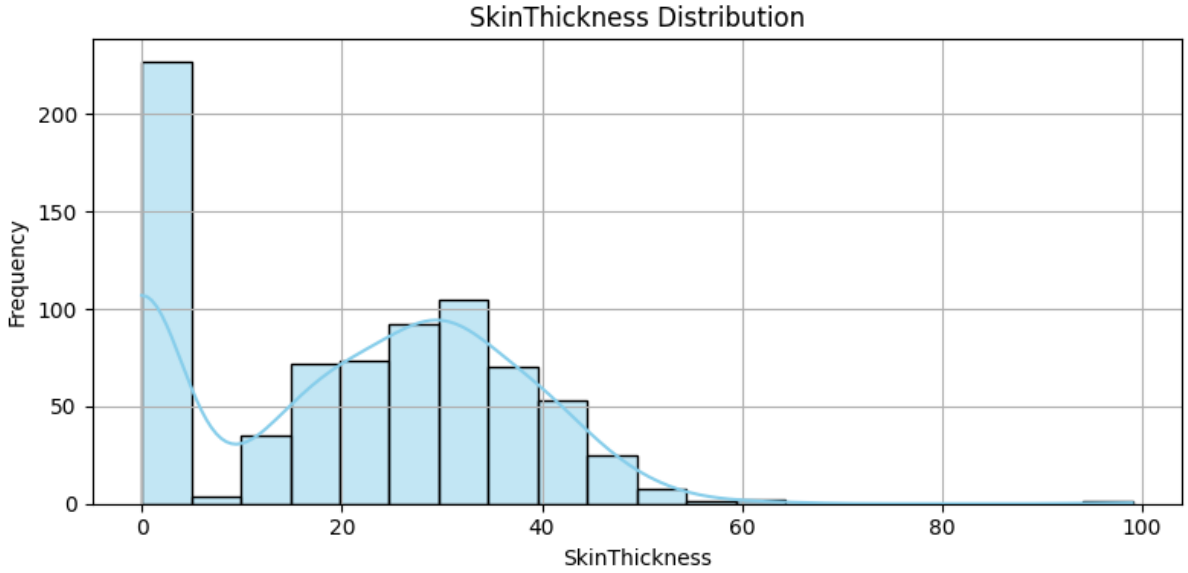
```python
# === Model Training ===
X = df.drop('Outcome', axis=1)
y = df['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42

start_train = time.time()
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
end_train = time.time()

start_pred = time.time()
y_pred = model.predict(X_test)
end_pred = time.time()

# === Evaluation ===
accuracy = accuracy_score(y_test, y_pred)
cm = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"✅ Model Accuracy: {accuracy:.2f}")
print(f"⏱️ Training time: {end_train - start_train:.4f} seconds")
print(f"⏱️ Prediction time: {end_pred - start_pred:.4f} seconds")
print("\n📄 Classification Report:")
print(report)

# === Confusion Matrix Plot ===
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=model.classes_)
disp.plot(cmap='Blues', values_format='d')
plt.title(f'Confusion Matrix (Accuracy: {accuracy:.2f})')
plt.grid(False)
plt.tight_layout()
plt.show()
```

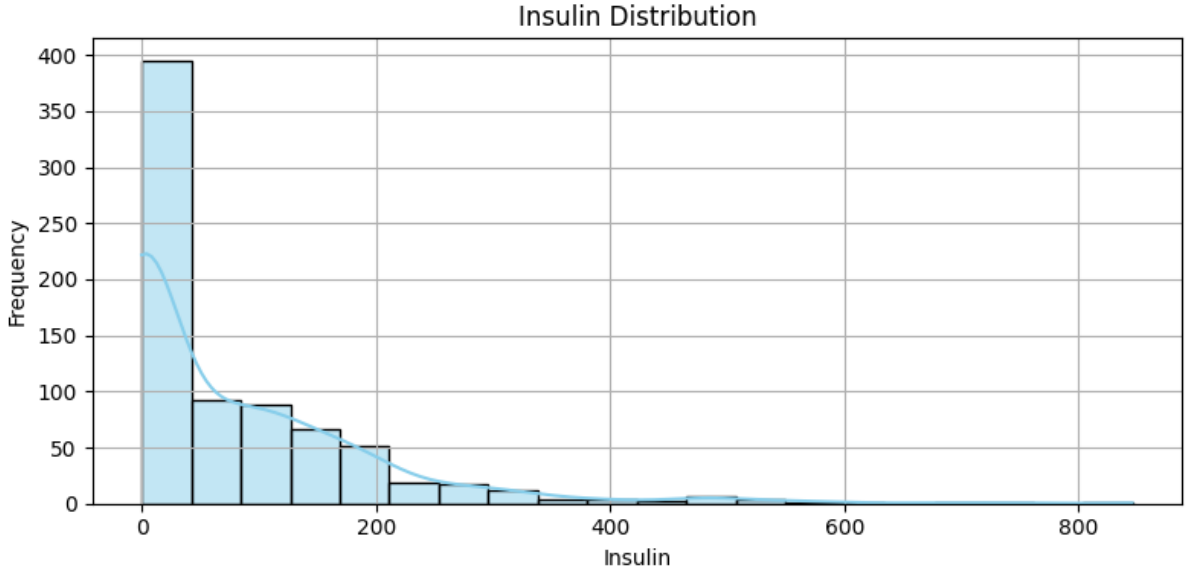
```python
# === Feature Importance - Horizontal Bar Chart ===
importances = model.feature_importances_
feat_imp_df = pd.DataFrame({'Feature': X.columns, 'Importance': importances})
feat_imp_df = feat_imp_df.sort_values(by='Importance', ascending=True)  # For horizontal chart

plt.figure(figsize=(10, 6))
plt.barh(feat_imp_df['Feature'], feat_imp_df['Importance'], color='blue')  # Horizontal bars
plt.xlabel('Importance')
plt.ylabel('Feature')
plt.title('Feature Importance - Random Forest')
plt.tight_layout()
plt.show()
```

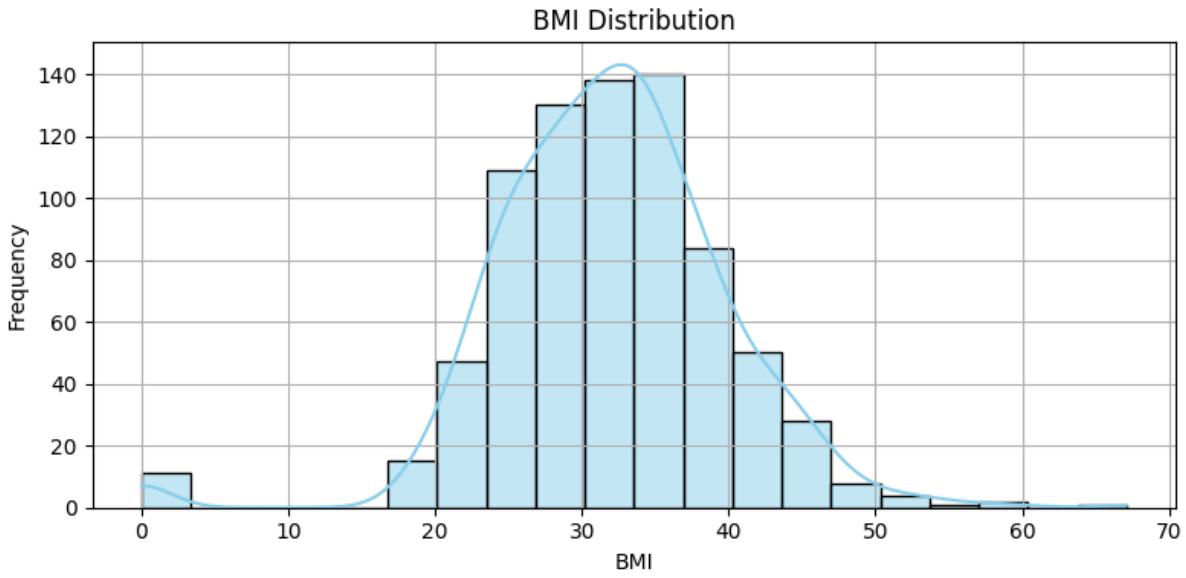


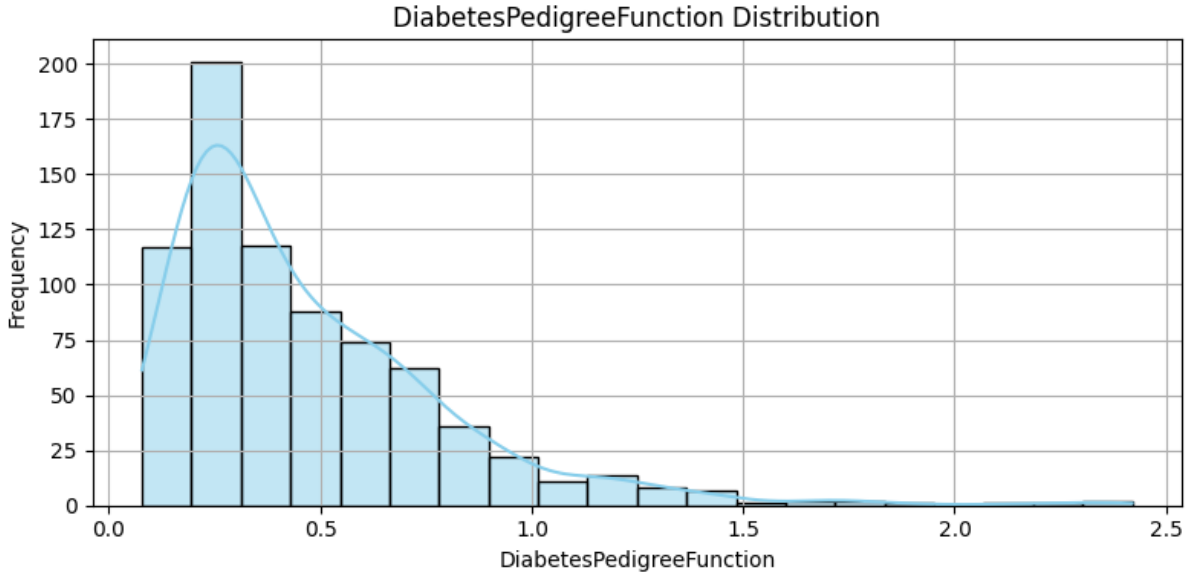


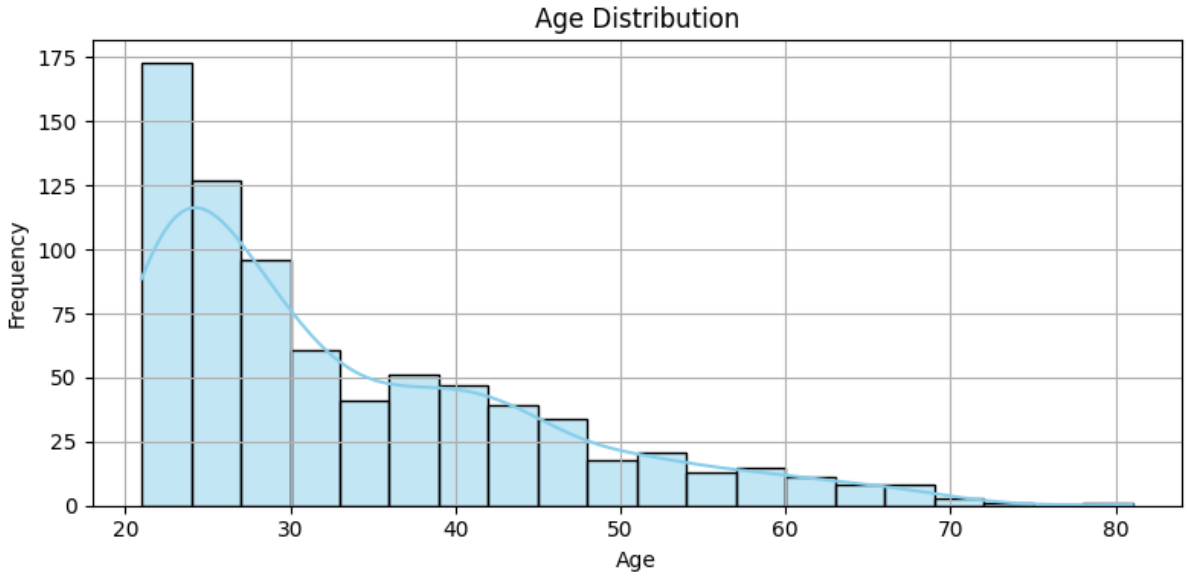

BloodPressure Distribution
Frequency
BloodPressure
0
20
40
60
80
100
120
0
20
40
60
80
100
120
140

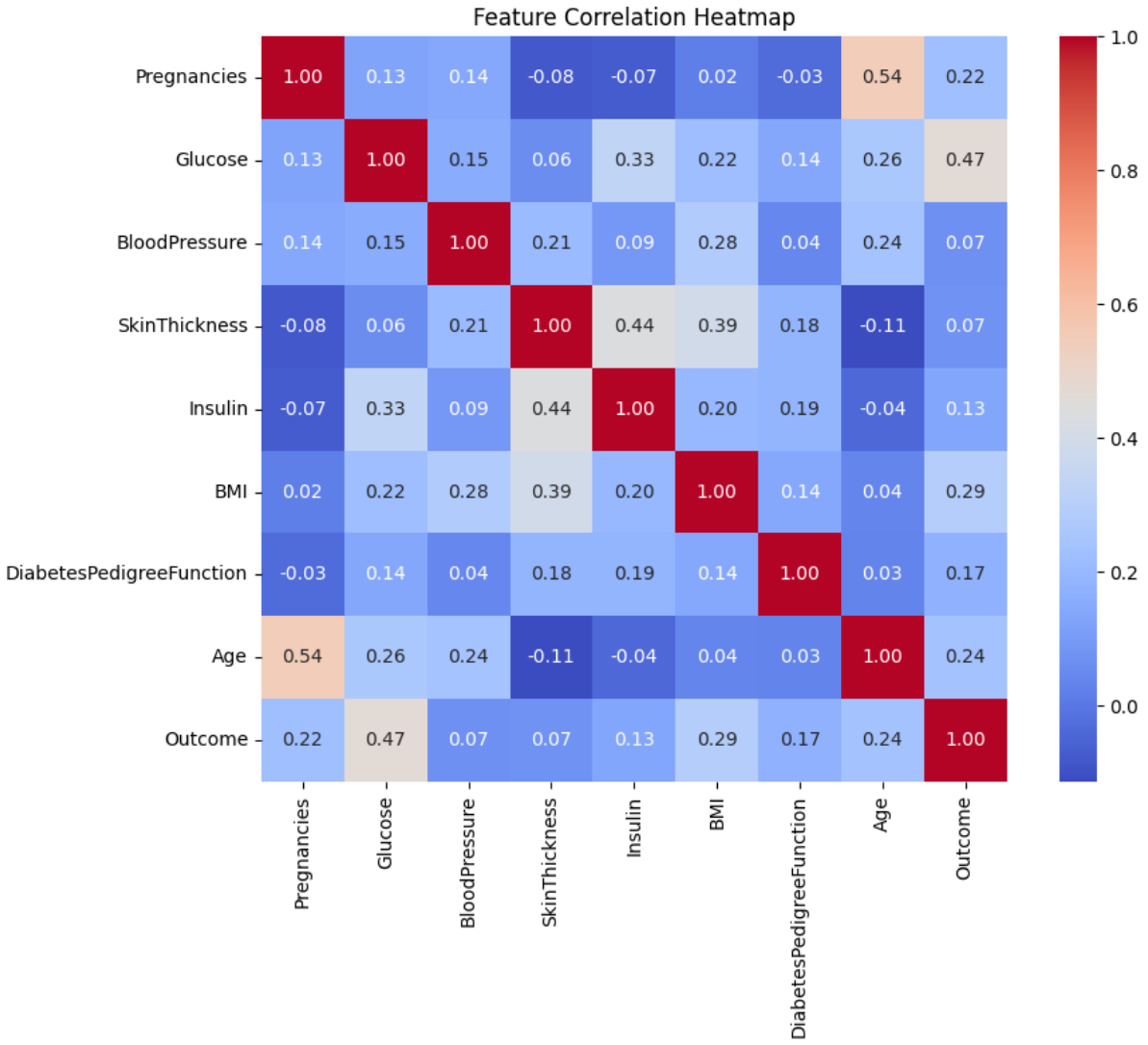

SkinThickness Distribution
Frequency
SkinThickness
0
20
40
60
80
100
0
50
100
150
200


Insulin Distribution
Frequency
Insulin
0
200
400
600
800
0
50
100
150
200
250
300
350
400


BMI Distribution
Frequency
BMI
0
10
20
30
40
50
60
70
0
20
40
60
80
100
120
140


DiabetesPedigreeFunction Distribution
Frequency
DiabetesPedigreeFunction
0.0
0.5
1.0
1.5
2.0
2.5
0
25
50
75
100
125
150
175
200


Age Distribution
Frequency
Age
20
30
40
50
60
70
80
0
25
50
75
100
125
150
175


Feature Correlation Heatmap
Pregnancies 1.00 0.13 0.14 -0.08 -0.07 0.02 -0.03 0.54 0.22
Glucose 0.13 1.00 0.15 0.06 0.33 0.22 0.14 0.26 0.47
BloodPressure 0.14 0.15 1.00 0.21 0.09 0.28 0.04 0.24 0.07
SkinThickness -0.08 0.06 0.21 1.00 0.44 0.39 0.18 -0.11 0.07
Insulin -0.07 0.33 0.09 0.44 1.00 0.20 0.19 -0.04 0.13
BMI 0.02 0.22 0.28 0.39 0.20 1.00 0.14 0.04 0.29
DiabetesPedigreeFunction -0.03 0.14 0.04 0.18 0.19 0.14 1.00 0.03 0.17
Age 0.54 0.26 0.24 -0.11 -0.04 0.04 0.03 1.00 0.24
Outcome 0.22 0.47 0.07 0.07 0.13 0.29 0.17 0.24 1.00
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1.0
0.8
0.6
0.4
0.2
0.0

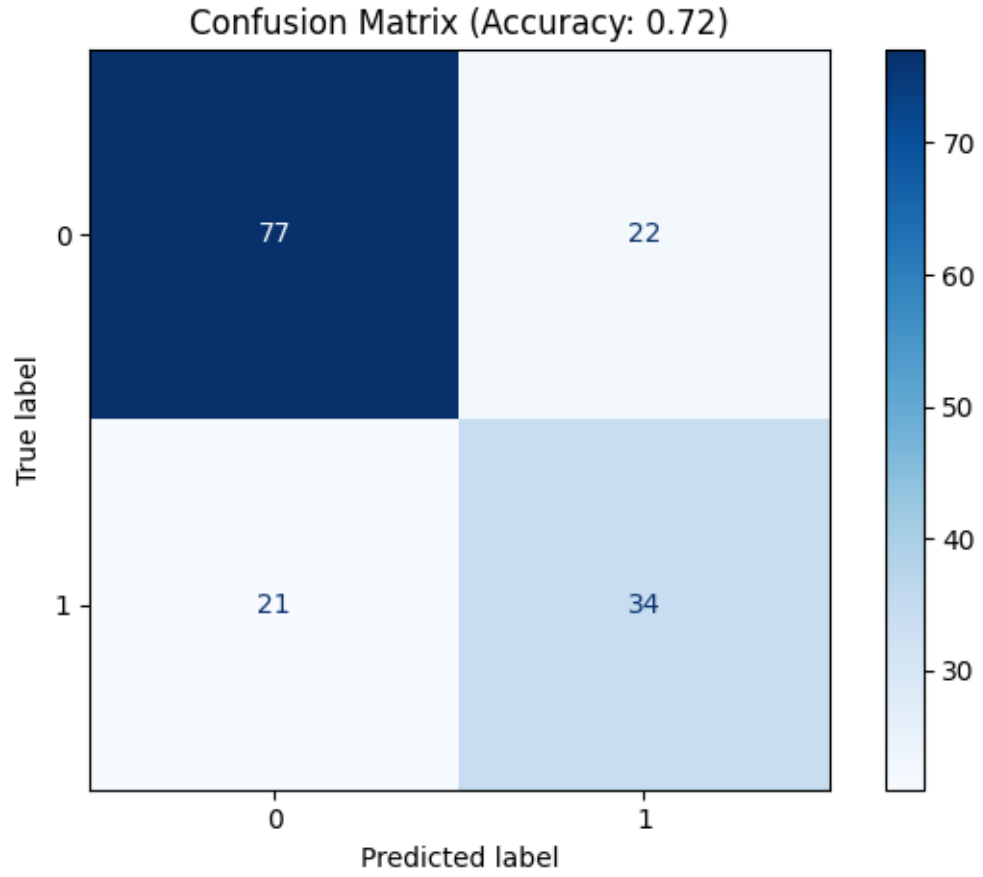
✅ Model Accuracy: 0.72

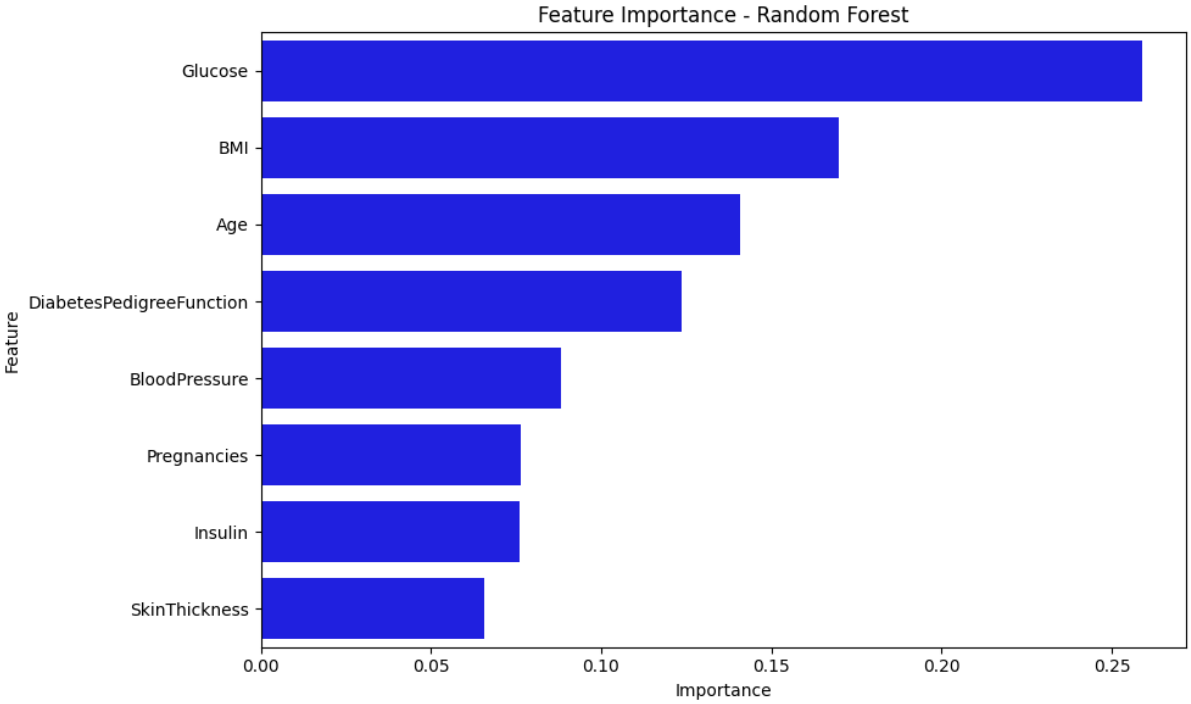
⏱️ Training time: 0.2410 seconds

⏱️ Prediction time: 0.0106 seconds

📄 Classification Report:

	precision	recall	f1-score	support
0	0.79	0.78	0.78	99
1	0.61	0.62	0.61	55
accuracy			0.72	154
macro avg	0.70	0.70	0.70	154
weighted avg	0.72	0.72	0.72	154




Feature Importance - Random Forest
Glucose
BMI
Age
DiabetesPedigreeFunction
BloodPressure
Pregnancies
Insulin
SkinThickness
Feature
0.00
0.05
0.10
0.15
0.20
0.25
Importance