A

PROJECT ON

DIABETIC PREDICTION (MACHINE LEARNING)

SUBMITTED BY
Ms. Smriti Singh (24558)

SUBMITTED TO
SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE

IN PARTIAL FULFILLMENT OF DEGREE
MASTERS OF COMPUTER APPLICATION (SEM- 1)

UNDER THE GUIDANCE OF
Ms. DEEPALI GAVHANE

Through,

SVIMS
Sadhu Vaswani Institute
of Management Studies
BELIEVE AND ACHIEVE

Sadhu Vaswani Institute of Management Studies For Girls,
Koregaon Park, Pune-411001 2024-2025

Sadhu Vaswani Institute of Management Studies for Girls

ACADEMIC YEAR 2024-26

MCA I – SEMESTER II

Abstract

Diabetes has become a major global health concern, affecting millions of people. Early detection and intervention can help manage the condition and reduce associated risks. This project uses machine learning to predict the likelihood of diabetes in individuals, based on various health-related parameters. We utilized the Pima Indians Diabetes dataset, a widely used resource for diabetes research. The dataset contains features such as age, BMI, glucose levels, and insulin, among others. A Random Forest model was trained to classify individuals as either diabetic or non-diabetic. The model was then deployed into a web application using Streamlit, providing an interactive platform where users can input their medical data and receive real-time predictions. This project demonstrates how machine learning can be applied to healthcare, improving decision-making and enabling early diagnosis of diabetes.

Objective

The primary goal of this project is to create a machine learning-based prediction model for diabetes using a dataset of medical records. This model predicts whether an individual is likely to have diabetes based on factors such as age, BMI, and glucose levels. The secondary goal is to integrate this model into a user-friendly web application using Streamlit, which allows users to input their medical data and receive real-time predictions. This system aims to assist healthcare professionals and individuals in the early detection of diabetes, promoting awareness and enabling timely interventions to prevent the progression of the disease.

Introduction

Diabetes is a chronic medical condition characterized by elevated blood sugar levels. It is one of the leading causes of death worldwide, contributing to complications such as heart disease, kidney failure, and nerve damage. Early prediction and diagnosis of diabetes are critical for effective management and lifestyle modifications. The emergence of machine learning has transformed healthcare by allowing for the analysis of large datasets to detect patterns and make accurate predictions. Machine learning models, such as Random Forest, are capable of processing complex datasets and making predictions based on multiple factors. This project aims to predict the likelihood of diabetes in individuals using a machine learning model and provide an interactive web application for easy access to the model's predictions.

Tools and Technologies Used

- **Programming Language:** Python
  Python is chosen due to its versatility, ease of use, and extensive library support for data science and machine learning.
- **Libraries:**
  - **Pandas:** Utilized for data manipulation, cleaning, and preprocessing.
  - **NumPy:** Used for numerical operations, such as handling missing values and scaling the data.
  - **Scikit-learn:** The primary library for machine learning, including algorithms for classification, model evaluation, and data splitting.
  - **Seaborn and Matplotlib:** Used for data visualization, including histograms, boxplots, and correlation heatmaps.
  - **Pickle:** Used to serialize and save the trained model and scaler for deployment.
  - **Streamlit:** A framework to create interactive, user-friendly web applications for machine learning models.
  - **Database**: SQLite

Model Development

- **Models Tested**:
  - Logistic Regression
  - Decision Tree
  - Random Forest (Best performer)

- **Model Used:** Random Forest Classifier
  The Random Forest model is selected due to its ability to handle complex datasets with high accuracy and its robustness.
- **Web Framework:** Streamlit
  Streamlit is used to build a simple and interactive web application that allows users to input their data and receive real-time predictions from the trained model.
- **IDE:** Jupyter Notebook for model development and Visual studio code for web app development.

Dataset Description

The dataset used for this project is the **Pima Indians Diabetes Dataset**, sourced from Kaggle. This dataset contains medical data of 768 individuals, each with 8 input features and a target output indicating whether or not the individual has diabetes.

Dataset Features:

Feature	**Description**
Pregnancies	Number of times pregnant
Glucose	Plasma glucose concentration (mg/dL)
Blood Pressure	Diastolic blood pressure (mm Hg)
Skin Thickness	Triceps skin fold thickness (mm)
Insulin	2-Hour serum insulin (mu U/ml)
BMI	Body Mass Index (kg/m^2)
Diabetes Pedigree Function	Diabetes pedigree function
Age	Age in years
Outcome	Class variable (0: No diabetes, 1: Diabetes)

Data Preprocessing:

- Missing values were handled by replacing zero values in certain columns with the mean of those columns.
- Standardization was performed using StandardScaler to normalize the features, improving the model's performance and ensuring that features are on a similar scale.

Machine Learning Workflow

Step 1: Data Collection and Import

The dataset is downloaded from Kaggle and imported using the pandas library. It is then examined for any missing or inconsistent data. Basic exploration is performed to understand the data structure and features.

```
import pandas as pd
```

```
data = pd.read_csv('diabetes.csv')

data.head()
```

Step 2: Data Preprocessing

We replaced zero values in columns like glucose, blood pressure, and BMI with the mean of those columns, as zero is not a valid value for those features.

```
data.replace(0, data.mean(), inplace=True)
```

Step 3: Feature Selection and Splitting

All 8 features are used for prediction, and the dataset is split into training (80%) and testing (20%) sets. This ensures the model is evaluated on unseen data.

```
from sklearn.model_selection import train_test_split

X = data.drop('Outcome', axis=1)

y = data['Outcome']

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

Step 4: Model Training

We chose the **Random Forest Classifier** due to its ability to handle complex datasets with high accuracy. The model is trained on the training dataset.

```
from sklearn.ensemble import RandomForestClassifier

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)
```

Step 5: Model Evaluation

The model is evaluated using accuracy score, confusion matrix, and classification report. We achieved approximately 77% accuracy with the model.

```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy}")

print(confusion_matrix(y_test, y_pred))

print(classification_report(y_test, y_pred))
```

Step 6: Model Deployment

After training, the model is serialized and saved using pickle, making it ready for deployment.

```
import pickle

with open('diabetes_model.pkl', 'wb') as f:

  pickle.dump(model, f)
```

Streamlit Web Application

Page Setup and Layout

The Streamlit app starts with a title and header, followed by an input form where users can input their medical data. We use Streamlit’s built-in functions for creating interactive widgets.

```
import streamlit as st

st.title("Diabetes Prediction App")
```

Input Form

Users input values for age, BMI, glucose level, etc., using Streamlit's number_input widgets.

```
glucose = st.number_input("Glucose Level", min_value=0)

bmi = st.number_input("BMI", min_value=0)

age = st.number_input("Age", min_value=0)
```

Prediction Logic and Display

Once the user submits the data, the inputs are passed through the model, and the prediction is displayed.

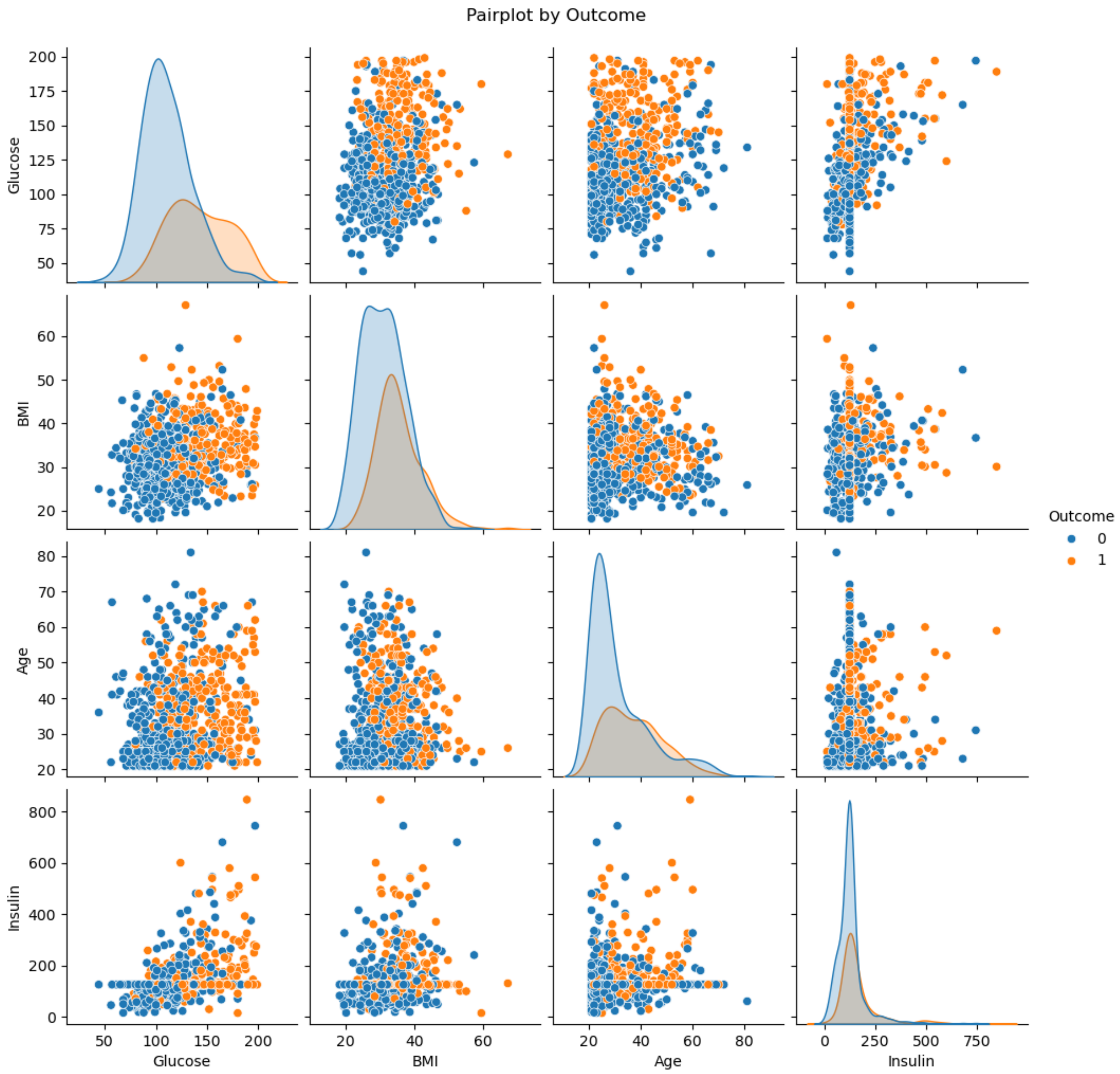
```
import pickle

with open('diabetes_model.pkl', 'rb') as f:

  model = pickle.load(f)


scaled_data = scaler.transform([[glucose, bmi, age]])  # scale data

prediction = model.predict(scaled_data)
```

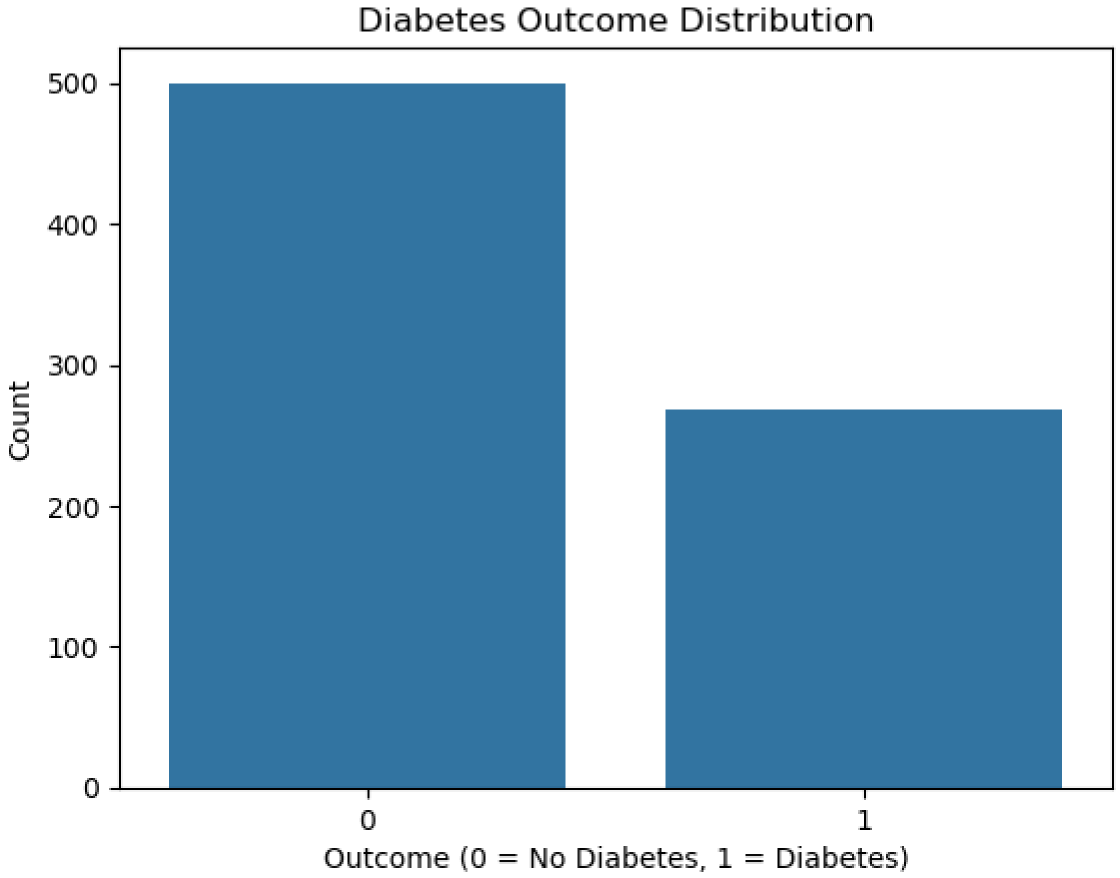
Database Integration (SQLite)

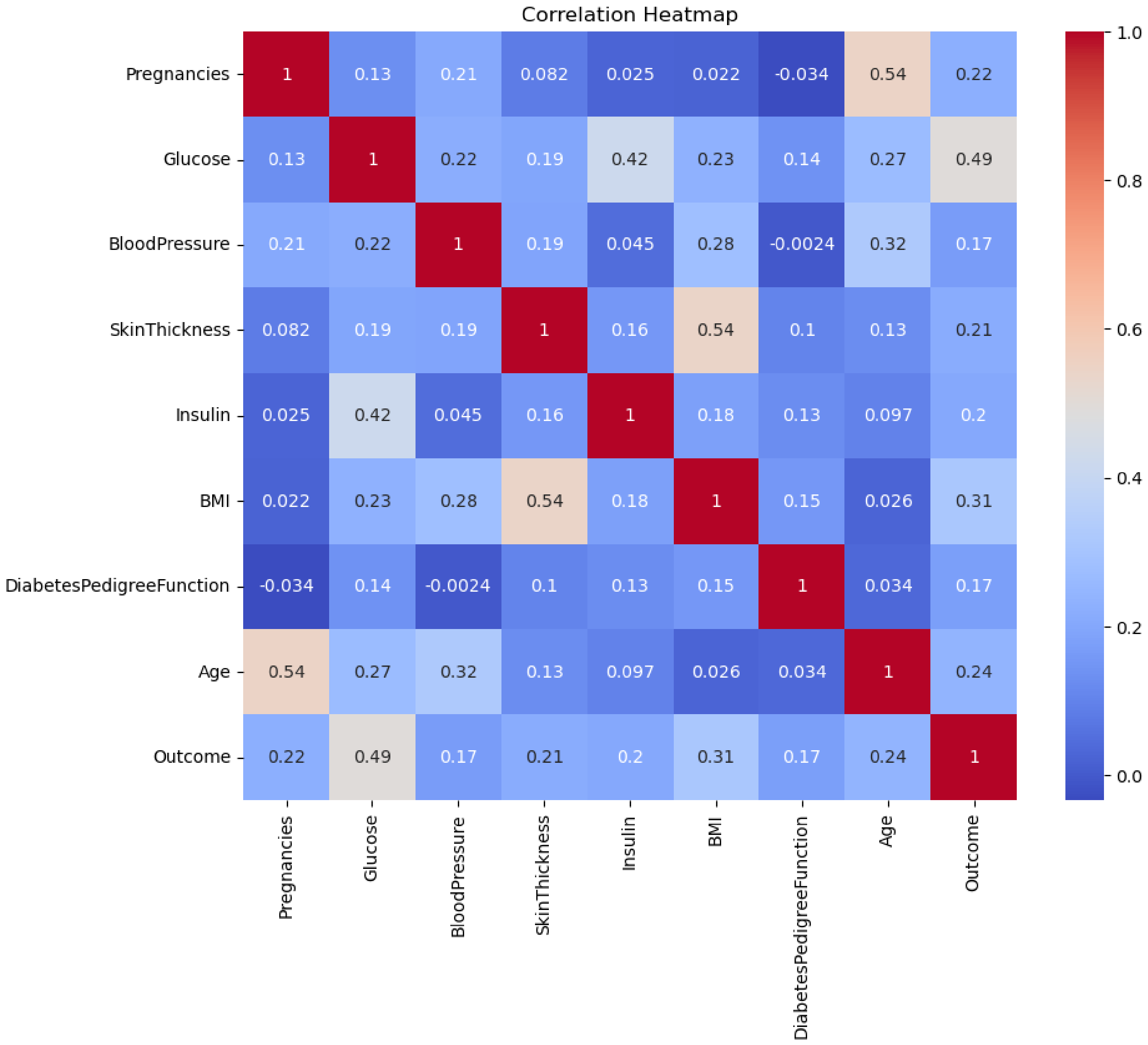
- Each prediction, along with input values, is **stored in a SQLite database** named diabetes_predictions.db.
- Users can view saved predictions using a toggle checkbox.
- Schema:

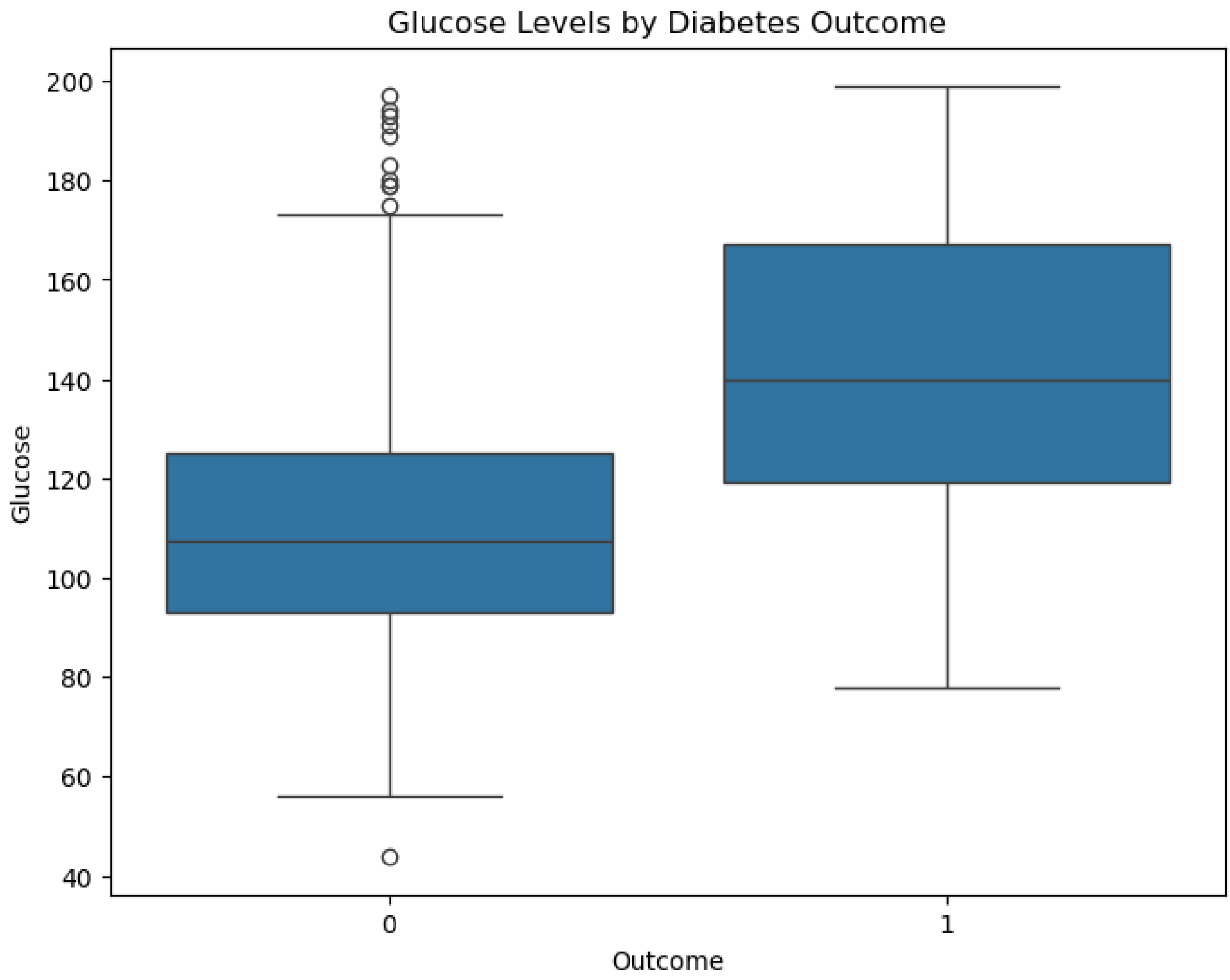
```
CREATE TABLE IF NOT EXISTS predictions (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  pregnancies INTEGER,
  glucose INTEGER,
  blood_pressure INTEGER,
  skin_thickness INTEGER,
  insulin INTEGER,
  bmi REAL,
  dpf REAL,
  age INTEGER,
  result INTEGER
);
```

Data Visualizations and Insights


Pairplot by Outcome
Glucose
BMI
Age
Insulin
200
175
150
125
100
75
50
60
50
40
30
20
80
70
60
50
40
30
20
800
600
400
200
0
50
100
150
200
Glucose
20
40
60
BMI
20
40
60
80
Age
0
250
500
750
Insulin
Outcome
● 0
● 1


Diabetes Outcome Distribution
Count
500
400
300
200
100
0
0
1
Outcome (0 = No Diabetes, 1 = Diabetes)


Correlation Heatmap
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1 0.13 0.21 0.082 0.025 0.022 -0.034 0.54 0.22
0.13 1 0.22 0.19 0.42 0.23 0.14 0.27 0.49
0.21 0.22 1 0.19 0.045 0.28 -0.0024 0.32 0.17
0.082 0.19 0.19 1 0.16 0.54 0.1 0.13 0.21
0.025 0.42 0.045 0.16 1 0.18 0.13 0.097 0.2
0.022 0.23 0.28 0.54 0.18 1 0.15 0.026 0.31
-0.034 0.14 -0.0024 0.1 0.13 0.15 1 0.034 0.17
0.54 0.27 0.32 0.13 0.097 0.026 0.034 1 0.24
0.22 0.49 0.17 0.21 0.2 0.31 0.17 0.24 1
1.0
0.8
0.6
0.4
0.2
0.0


Glucose Levels by Diabetes Outcome
Glucose
200
180
160
140
120
100
80
60
40
0
1
Outcome

Results and Example Predictions

Example 1: Child (No Diabetes Risk)

- **Age:** 10
- **Pregnancies:** 0
- **Glucose:** 85
- **Blood Pressure:** 70
- **Skin Thickness:** 20
- **Insulin:** 80
- **BMI:** 18.5
- **Diabetes Pedigree Function:** 0.1

Prediction Output:
Result: No diabetes (0)

Example 2: Adult (Diabetes Risk)

- **Age:** 35
- **Pregnancies:** 1
- **Glucose:** 140
- **Blood Pressure:** 80
- **Skin Thickness:** 25
- **Insulin:** 150
- **BMI:** 28
- **Diabetes Pedigree Function:** 0.5

Prediction Output:
Result: Diabetes risk (1)

Example 3: Elderly Pregnant Woman (Diabetes Risk)

- **Age:** 65
- **Pregnancies:** 3
- **Glucose:** 190
- **Blood Pressure:** 90
- **Skin Thickness:** 30

- **Insulin:** 145
- **BMI:** 32
- **Diabetes Pedigree Function:** 0.9

Prediction Output:
Result: Diabetes risk (1)

Challenges Faced

- **Database locked error** due to concurrent writes in SQLite — resolved by using engine.begin() for safe transactions.
- Ensuring smooth scaling and deployment across environments.

Conclusion

This project successfully demonstrates a real-world application of machine learning in the healthcare domain. The system offers:

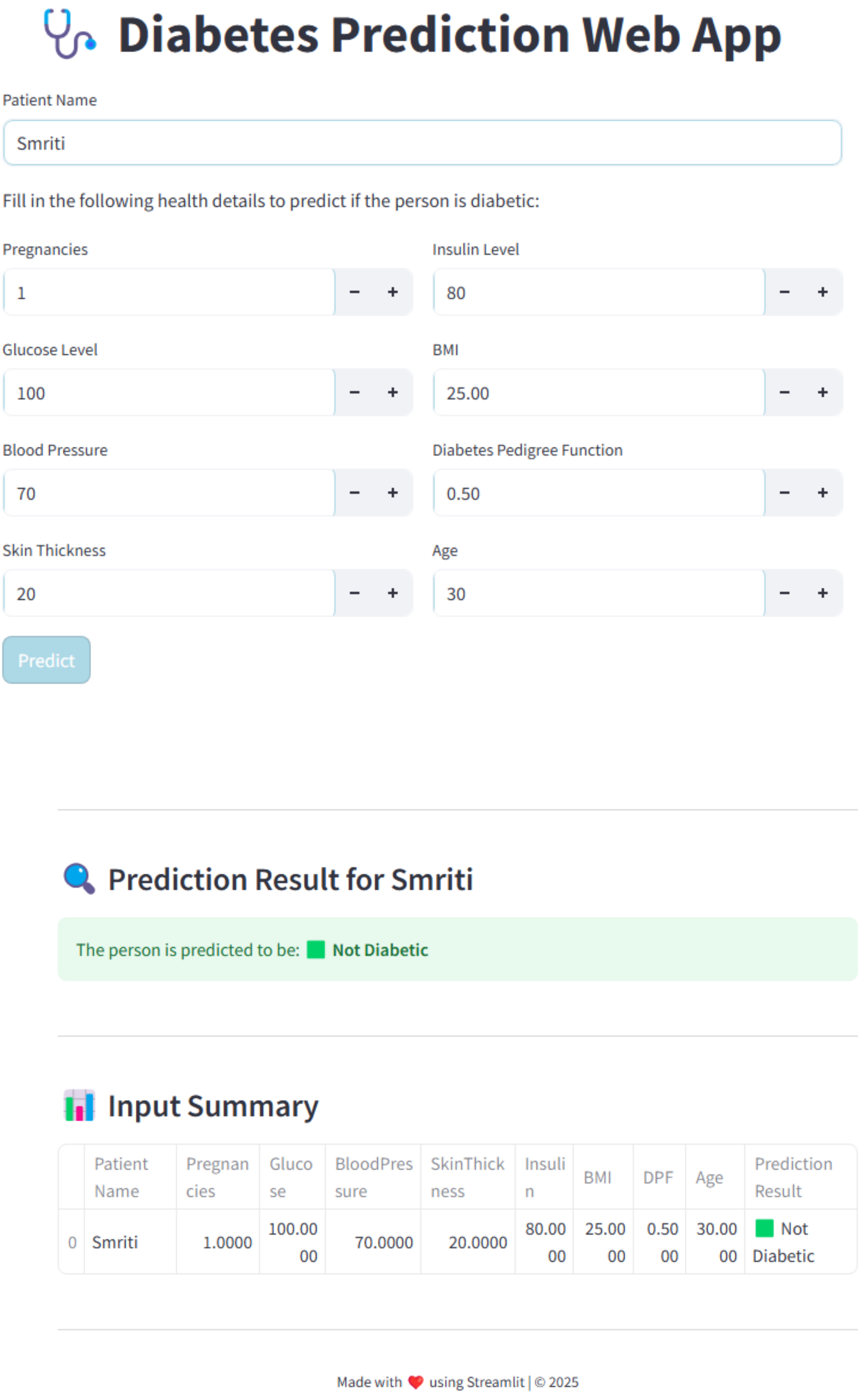
- Accurate prediction
- Easy-to-use web interface
- Data persistence for analysis

It can be further improved by:

- Deploying to cloud (Heroku, AWS)
- Using advanced deep learning models
- Adding user authentication and role-based access

Screenshots

(Screenshots of the Streamlit UI, prediction output, and database view)



🩺 Diabetes Prediction Web App

Patient Name

Smriti

Fill in the following health details to predict if the person is diabetic:

Pregnancies

1

Insulin Level

80

Glucose Level

100

BMI

25.00

Blood Pressure

70

Diabetes Pedigree Function

0.50

Skin Thickness

20

Age

30

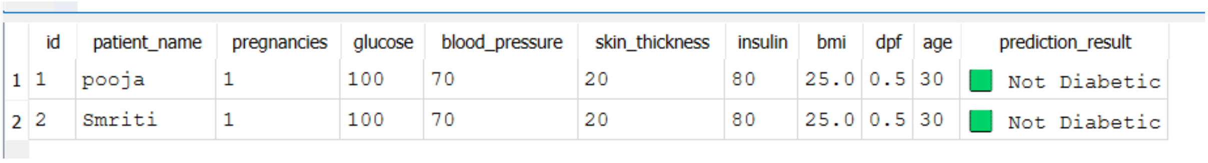
🔍 Prediction Result for Smriti

The person is predicted to be: 🟩 **Not Diabetic**

📊 Input Summary

	Patient Name	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Prediction Result
0	Smriti	1.0000	100.0000	70.0000	20.0000	80.0000	25.0000	0.5000	30.0000	🟩 Not Diabetic

Made with ❤️ using Streamlit | © 2025



	id	patient_name	pregnancies	glucose	blood_pressure	skin_thickness	insulin	bmi	dpf	age	prediction_result
1	1	pooja	1	100	70	20	80	25.0	0.5	30	🟩 Not Diabetic
2	2	Smriti	1	100	70	20	80	25.0	0.5	30	🟩 Not Diabetic

Limitations of the Project

While the **Diabetes Prediction using Machine Learning and Streamlit Web Application** project showcases a promising approach to early diabetes detection, there are several limitations and areas where the project can be improved. These limitations can help provide a better understanding of the model's current performance and potential shortcomings in real-world applications.

1. **Limited Dataset**
   - The project relies on the **Pima Indians Diabetes Dataset**, which may not fully represent the diversity of populations globally. The dataset primarily contains information about individuals from a specific ethnic group (Pima Indians), and hence, the model may not generalize well to other populations with different demographic characteristics.
   - **Solution:** Incorporating more diverse datasets and ensuring a broader range of medical features can improve the model's robustness and generalizability.
2. **Feature Limitations**
   - The model only uses 8 features to predict diabetes. Some important factors such as **HbA1c levels**, **genetic predisposition**, and **lifestyle data (e.g., diet, physical activity)** are not considered.
   - **Solution:** Including additional features, such as lifestyle choices, family history, and more detailed blood tests, could potentially improve the model's accuracy and predictive power.
3. **Class Imbalance**
   - The dataset exhibits an imbalance between the two classes (diabetic vs non-diabetic), with more individuals not having diabetes (Outcome = 0) than those who do (Outcome = 1). This imbalance can lead to biased predictions where the model may favor the majority class (non-diabetic).

- **Solution:** Techniques like **oversampling**, **undersampling**, or using **class weights** in the model could help address this imbalance and improve the model's ability to predict the minority class (diabetic cases).

4. **Overfitting Risk**
   - The Random Forest model, while robust, has the potential for **overfitting** if the number of trees or other hyperparameters are not properly tuned. Overfitting can result in high accuracy on the training set but poor performance on unseen data.
   - **Solution:** Proper cross-validation, hyperparameter tuning, and model regularization techniques can help mitigate the risk of overfitting.
5. **Lack of Real-Time Data Integration**
   - The model currently works with static data, and while it can predict based on user input, it doesn't integrate real-time medical data from wearable devices or sensors that could provide continuous monitoring of relevant metrics like glucose levels, heart rate, or insulin levels.
   - **Solution:** Incorporating real-time data from wearable devices, such as **continuous glucose monitors** or **smartwatches**, could make the model more dynamic and provide better predictive capabilities.
6. **Accuracy and Predictive Power**
   - With an accuracy of approximately **77%**, the model's performance may not be sufficient for a clinical setting, where higher accuracy and precision are critical. The model might produce false positives (predicting diabetes when the patient does not have it) or false negatives (failing to detect diabetes in an individual who actually has it).
   - **Solution:** Testing with more advanced models (e.g., **XGBoost**, **Support Vector Machines**) or exploring ensemble methods could help improve predictive accuracy.
7. **No Consideration for Temporal Data**
   - Diabetes is a progressive disease, and its likelihood may change over time based on an individual's lifestyle and medical history. The model does not account for temporal data, meaning it treats each prediction as an isolated event without considering any long-term changes in an individual's health status.

   - **Solution:** Implementing **time-series models** that track changes in key health indicators over time could enhance the model's predictive capability.

8. **No Model Explanation (Black Box)**
   - The Random Forest model, like many machine learning models, operates as a "black box," meaning it does not offer a transparent explanation of how it arrives at a particular decision. In healthcare applications, understanding why a model makes a certain prediction is critical for trust and actionability.
   - **Solution:** Utilizing **explainable AI (XAI)** techniques such as **SHAP (Shapley Additive Explanations)** or **LIME (Local Interpretable Model-Agnostic Explanations)** could help provide more insight into the model's decision-making process.

9. **Potential Security and Privacy Concerns**
   - The web application requires user input of sensitive health data, which raises concerns about the **security and privacy** of personal information. Without proper encryption or secure data handling practices, there could be risks of data breaches.
   - **Solution:** Implementing **data encryption**, **secure HTTPS connections**, and ensuring **compliance with privacy regulations** like GDPR and HIPAA is essential to protect users' health information.

10. **Lack of Clinical Validation**

- Although the model provides predictions based on historical data, it has not undergone clinical trials or validation with real-world data from healthcare institutions. This means that while the model may work in controlled environments, it may not be fully reliable in clinical practice.
- **Solution:** Collaboration with healthcare professionals and clinical trials to validate the model's effectiveness in real-world settings could enhance its credibility and reliability.

Conclusion on Limitations

While this project demonstrates a useful tool for diabetes prediction using machine learning, it is important to recognize its limitations. Addressing these challenges, such as improving model generalization, increasing accuracy, and integrating additional data, can enhance the system's usability and effectiveness in healthcare settings. Future work may involve incorporating more complex models, utilizing real-time health data, and ensuring better explainability to make the system more robust and applicable to real-world medical use.