Diabetes Prediction Web App using Machine Learning

Aim:

To predict the likelihood of diabetes by analyzing various health-related features such as glucose level, BMI, insulin, and more using classification techniques like logistic regression.

Programs:

Model.py

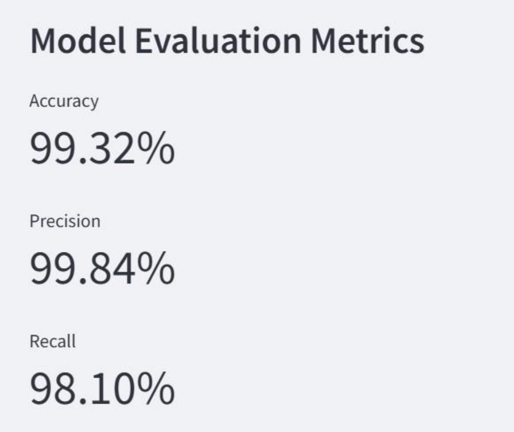
```
import pandas as pd
import numpy as np
import joblib
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score

# Load dataset
df = pd.read_csv("D:/ml_assignment/diabetes_dataset.csv")

# Drop unnecessary columns
columns_to_remove = ["LDL", "Hip Circumference", "WHR", "Diet Type"]
existing_columns = [col for col in columns_to_remove if col in df.columns]
X = df.drop(columns=["Outcome"] + existing_columns)
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)

# Scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Model
model = LogisticRegression()
model.fit(X_train_scaled, y_train)

# Metrics
y_pred = model.predict(X_test_scaled)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred, zero_division=0))
print("Recall:", recall_score(y_test, y_pred, zero_division=0))

# Save model and scaler
joblib.dump(model, "logistic_model.pkl")
joblib.dump(scaler, "scaler.pkl")
joblib.dump(X.columns.tolist(), "features.pkl")
```

app.py

```
import streamlit as st
import joblib
import numpy as np

# Load saved components
model = joblib.load("logistic_model.pkl")
scaler = joblib.load("scaler.pkl")
features = joblib.load("features.pkl")

# Streamlit config
st.set_page_config(page_title="Diabetes Prediction App", page_icon="□", layout="centered")
st.markdown("""
    <style>
        .stApp { background-color: #f0f2f6; }
        .title {
            text-align: center;
            font-size: 36px;
            font-weight: bold;
            color: #4CAF50;
        }
        .stButton>button {
            background-color: #4CAF50;
            color: white;
            font-size: 20px;
            border-radius: 8px;
            padding: 10px;
        }
    </style>
""", unsafe_allow_html=True)

# Title and instructions
st.markdown("<p class='title'>Diabetes Prediction App □</p>", unsafe_allow_html=True)
st.write("Fill in the details below to predict your diabetes risk.")

# Input form
st.subheader("Enter Your Health Information")
input_data = []
col1, col2 = st.columns(2)

for i, col in enumerate(features):
    with col1 if i % 2 == 0 else col2:
        value = st.number_input(col, value=0.0)
        input_data.append(value)

# Predict button
```

```
if st.button("🔍 Predict"):
    input_scaled = scaler.transform([input_data])
    prediction = "🛑 Diabetic" if model.predict(input_scaled)[0] == 1 else "✅ Not Diabetic"
    st.success(f"### Prediction: {prediction}")
```

output:



Model Evaluation Metrics

Accuracy

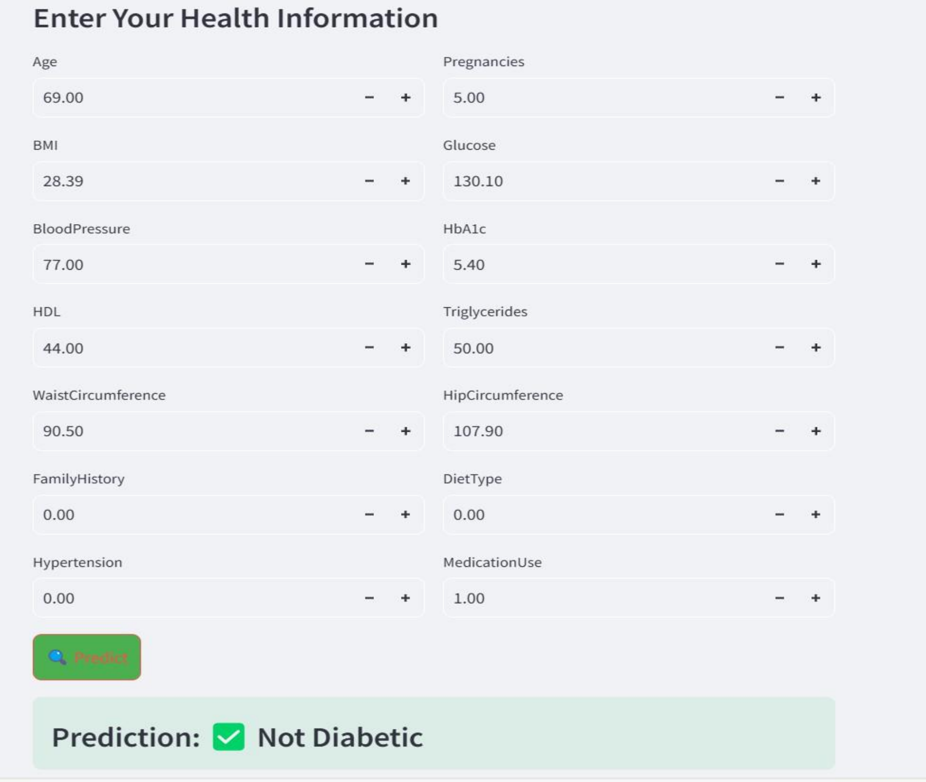
99.32%

Precision

99.84%

Recall

98.10%



Enter Your Health Information

Age 69.00	Pregnancies 5.00
BMI 28.39	Glucose 130.10
BloodPressure 77.00	HbA1c 5.40
HDL 44.00	Triglycerides 50.00
WaistCircumference 90.50	HipCircumference 107.90
FamilyHistory 0.00	DietType 0.00
Hypertension 0.00	MedicationUse 1.00

Prediction: ✅ Not Diabetic

Link:

https://www.kaggle.com/datasets/hasibur013/diabetes-dataset