'''NAME = Gajanan Santosh Purud

PROJECT NAME = Diabetes Checkup '''

#CODE

#pip install streamlit #pip install pandas

#pip install sklearn

# # IMPORT STATEMENTS

import streamlit as st import pandas as pd

from PIL import Image import numpy as np

import matplotlib.pyplot as plt import plotly.figure\_factory as ff

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestClassifier from sklearn.model\_selection import train\_test\_split import seaborn as sns

df = pd.read\_csv(r'C:\Users\Admin\Desktop\PythonProject\DiabetesDetection\(diabetes.csv))

# HEADINGS

st.title('Diabetes Checkup') st.sidebar.header('Patient Data') st.subheader('Training Data Stats') st.write(df.describe())

st.subheader('Visualisation')

st.bar\_chart(df)

# # X AND Y DATA

x = df.drop(['Outcome'], axis = 1) y = df.iloc[:, -1]

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x,y, test\_size = 0.2, random\_state

= 0)

# # FUNCTION

def user\_report():

pregnancies = st.sidebar.slider('Pregnancies', 0,17, 3 )

glucose = st.sidebar.slider('Glucose', 0,200, 120 ) bp = st.sidebar.slider('Blood Pressure', 0,122, 70 )

skinthickness = st.sidebar.slider('Skin Thickness', 0,100, 20 ) insulin = st.sidebar.slider('Insulin', 0,846, 79 )

bmi = st.sidebar.slider('BMI', 0,67, 20 )

dpf = st.sidebar.slider('Diabetes Pedigree Function', 0.0,2.4, 0.47 ) age = st.sidebar.slider('Age', 21,88, 33 )

user\_report\_data = { 'pregnancies':pregnancies, 'glucose':glucose, 'bp':bp,

'skinthickness':skinthickness, 'insulin':insulin,

'bmi':bmi,

'dpf':dpf,

'age':age

}

report\_data = pd.DataFrame(user\_report\_data, index=[0]) return report\_data

# # PATIENT DATA

user\_data = user\_report() st.subheader('Patient Data') st.write(user\_data)

# # MODEL

rf = RandomForestClassifier() rf.fit(x\_train, y\_train) user\_result = rf.predict(user\_data)

# # VISUALISATIONS

st.title('Visualised Patient Report')

# # COLOR FUNCTION

if user\_result[0]==0: color = 'blue'

else:

color = 'red'

# Age vs Pregnancies

st.header('Pregnancy count Graph (Others vs Yours)') fig\_preg = plt.figure()

ax1 = sns.scatterplot(x = 'Age', y = 'Pregnancies', data = df, hue = 'Outcome', palette

= 'Greens')

ax2 = sns.scatterplot(x = user\_data['age'], y = user\_data['pregnancies'], s = 150, color = color)

plt.xticks(np.arange(10,100,5)) plt.yticks(np.arange(0,20,2)) plt.title('0 - Healthy & 1 - Unhealthy') st.pyplot(fig\_preg)

# Age vs Glucose

st.header('Glucose Value Graph (Others vs Yours)') fig\_glucose = plt.figure()

ax3 = sns.scatterplot(x = 'Age', y = 'Glucose', data = df, hue = 'Outcome' , palette='magma')

ax4 = sns.scatterplot(x = user\_data['age'], y = user\_data['glucose'], s = 150, color = color)

plt.xticks(np.arange(10,100,5)) plt.yticks(np.arange(0,220,10)) plt.title('0 - Healthy & 1 - Unhealthy') st.pyplot(fig\_glucose)

# Age vs Bp

st.header('Blood Pressure Value Graph (Others vs Yours)') fig\_bp = plt.figure()

ax5 = sns.scatterplot(x = 'Age', y = 'BloodPressure', data = df, hue = 'Outcome', palette='Reds')

ax6 = sns.scatterplot(x = user\_data['age'], y = user\_data['bp'], s = 150, color = color)

plt.xticks(np.arange(10,100,5)) plt.yticks(np.arange(0,130,10)) plt.title('0 - Healthy & 1 - Unhealthy') st.pyplot(fig\_bp)

# Age vs St

st.header('Skin Thickness Value Graph (Others vs Yours)') fig\_st = plt.figure()

ax7 = sns.scatterplot(x = 'Age', y = 'SkinThickness', data = df, hue = 'Outcome', palette='Blues')

ax8 = sns.scatterplot(x = user\_data['age'], y = user\_data['skinthickness'], s = 150, color = color)

plt.xticks(np.arange(10,100,5)) plt.yticks(np.arange(0,110,10)) plt.title('0 - Healthy & 1 - Unhealthy') st.pyplot(fig\_st)

# Age vs Insulin

st.header('Insulin Value Graph (Others vs Yours)') fig\_i = plt.figure()

ax9 = sns.scatterplot(x = 'Age', y = 'Insulin', data = df, hue = 'Outcome', palette='rocket')

ax10 = sns.scatterplot(x = user\_data['age'], y = user\_data['insulin'], s = 150, color = color)

plt.xticks(np.arange(10,100,5)) plt.yticks(np.arange(0,900,50)) plt.title('0 - Healthy & 1 - Unhealthy') st.pyplot(fig\_i)

# Age vs BMI

st.header('BMI Value Graph (Others vs Yours)') fig\_bmi = plt.figure()

ax11 = sns.scatterplot(x = 'Age', y = 'BMI', data = df, hue = 'Outcome', palette='rainbow')

ax12 = sns.scatterplot(x = user\_data['age'], y = user\_data['bmi'], s = 150, color = color)

plt.xticks(np.arange(10,100,5)) plt.yticks(np.arange(0,70,5))

plt.title('0 - Healthy & 1 - Unhealthy') st.pyplot(fig\_bmi)

# Age vs Dpf

st.header('DPF Value Graph (Others vs Yours)') fig\_dpf = plt.figure()

ax13 = sns.scatterplot(x = 'Age', y = 'DiabetesPedigreeFunction', data = df, hue = 'Outcome', palette='YlOrBr')

ax14 = sns.scatterplot(x = user\_data['age'], y = user\_data['dpf'], s = 150, color = color)

plt.xticks(np.arange(10,100,5)) plt.yticks(np.arange(0,3,0.2)) plt.title('0 - Healthy & 1 - Unhealthy') st.pyplot(fig\_dpf)

# # OUTPUT

st.subheader('Your Report: ') output=''

if user\_result[0]==0:

output = 'You are not Diabetic' else:

output = 'You are Diabetic' st.title(output) st.subheader('Accuracy: ')

st.write(str(accuracy\_score(y\_test, rf.predict(x\_test))\*100)+'%')