

```
▶ import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Step 1: Generate synthetic healthcare dataset
def generate_health_data(samples=1000):
    np.random.seed(42)
    data = [
        {
            'Age': np.random.randint(20, 80, samples),
            'BloodPressure': np.random.randint(90, 180, samples),
            'Glucose': np.random.randint(70, 200, samples),
            'BMI': np.round(np.random.uniform(18, 40, samples), 2),
            'Cholesterol': np.random.randint(150, 300, samples),
            'Smoking': np.random.choice([0, 1], samples, p=[0.7, 0.3])
        }
    ]
    df = pd.DataFrame(data)

    # Define disease presence based on risk factor thresholds
    df['Disease'] = (
        (df['BloodPressure'] > 140) |
        (df['Glucose'] > 140) |
        (df['BMI'] > 30) |
        (df['Cholesterol'] > 240) |
        (df['Smoking'] == 1)
    ).astype(int)

    return df
```

```
# Step 2: Load and split the data
df = generate_health_data()
X = df.drop('Disease', axis=1)
y = df['Disease']

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

# Step 3: Train model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Step 4: Evaluate model
y_pred = model.predict(X_test)
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Step 5: Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='YlGnBu',
            xticklabels=['No Disease', 'Disease'],
            yticklabels=['No Disease', 'Disease'])
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Step 6: Predict on new patient input
def predict_patient_risk(age, bp, glucose, bmi, cholesterol, smoking):
    input_data = pd.DataFrame([[age, bp, glucose, bmi, cholesterol, smoking]],
                             columns=X.columns)
    prediction = model.predict(input_data)[0]
    return "Disease Detected" if prediction == 1 else "No Disease"
```

```
# Step 6: Predict on new patient input
def predict_patient_risk(age, bp, glucose, bmi, cholesterol, smoking):
    input_data = pd.DataFrame([[age, bp, glucose, bmi, cholesterol, smoking]],
                             columns=X.columns)
    prediction = model.predict(input_data)[0]
    return "Disease Detected" if prediction == 1 else "No Disease"

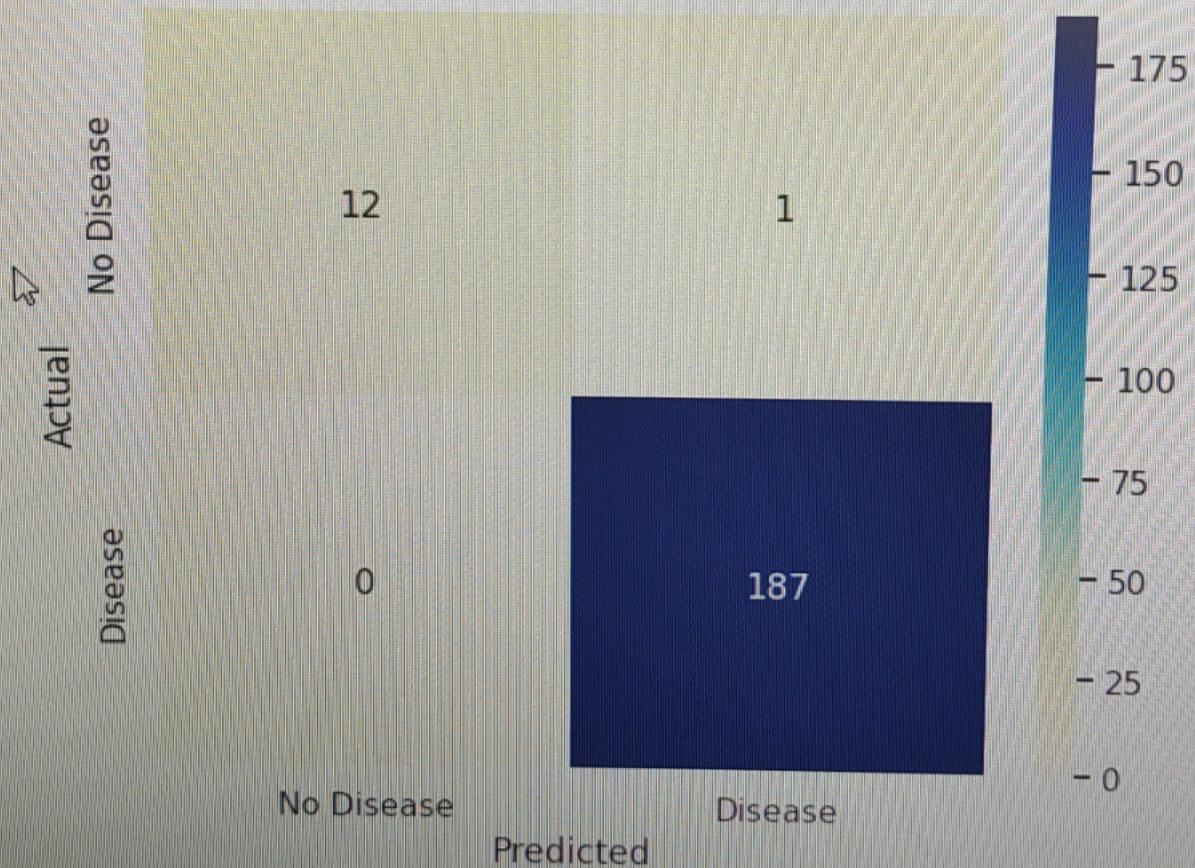
# Example prediction
result = predict_patient_risk(55, 145, 160, 31.5, 260, 1)
print("\nPrediction for New Patient: ", result)
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.92	0.96	13
1	0.99	1.00	1.00	187
accuracy			0.99	200
macro avg	1.00	0.96	0.98	200
weighted avg	1.00	0.99	0.99	200

	1	0.99	1.00	1.00	187
accuracy				0.99	200
macro avg		1.00	0.96	0.98	200
weighted avg		1.00	0.99	0.99	200

Confusion Matrix



Prediction for New Patient: Disease Detected