```
# File: disease_prediction.py

import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score

# Step 1: Generate Synthetic Medical Data
def generate_synthetic_data(num_samples=1000):
    np.random.seed(42)
    data = {
        'age': np.random.randint(18, 90, size=num_samples),
        'blood_pressure': np.random.randint(80, 180, size=num_samples),
        'cholesterol': np.random.randint(150, 300, size=num_samples),
        'smoker': np.random.choice([0, 1], size=num_samples),
        'diabetes': np.random.choice([0, 1], size=num_samples),
        'exercise_frequency': np.random.randint(0, 5, size=num_samples),
        'disease': np.random.choice([0, 1], size=num_samples, p=[0.7, 0.3]),  # Target variable
    }
    return pd.DataFrame(data)

# Step 2: Preprocess the Data
def preprocess_data(data):
    X = data.drop(columns=['disease'])  # Features
    y = data['disease']  # Target
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    return X_train, X_test, y_train, y_test

# Step 3: Train a Random Forest Classifier
def train_model(X_train, y_train):
    model = RandomForestClassifier(random_state=42, n_estimators=100)
    model.fit(X_train, y_train)
    return model

# Step 4: Evaluate the Model
def evaluate_model(model, X_test, y_test):
    predictions = model.predict(X_test)
    print("Classification Report:\n", classification_report(y_test, predictions))
    print("Accuracy Score:", accuracy_score(y_test, predictions))

# Main Function
if __name__ == "__main__":
    # Generate data
    medical_data = generate_synthetic_data()

    # Preprocess
    X_train, X_test, y_train, y_test = preprocess_data(medical_data)

    # Train model
    classifier = train_model(X_train, y_train)

    # Evaluate
    evaluate_model(classifier, X_test, y_test)
```

```
Classification Report:
                  precision    recall  f1-score   support

               0       0.70      0.94      0.80       143
               1       0.00      0.00      0.00        57

        accuracy                           0.67       200
       macro avg       0.35      0.47      0.40       200
    weighted avg       0.50      0.67      0.57       200

    Accuracy Score: 0.67
```