```
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
import joblib

# Step 1: Load or create sample patient dataset
def load_data():
    data = {
        'age': [25, 45, 50, 35, 29],
        'gender': [0, 1, 1, 0, 0],  # 0: Female, 1: Male
        'fever': [1, 0, 1, 0, 1],
        'cough': [1, 0, 1, 1, 1],
        'bp_high': [0, 1, 1, 0, 0],
        'diabetes': [0, 1, 1, 0, 0],
        'disease': [1, 1, 1, 0, 0]  # 1: Disease present, 0: Healthy
    }
    return pd.DataFrame(data)

# Step 2: Train model
def train_model(df):
    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)

    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
```

```
    predictions = model.predict(X_test)
    print("Accuracy:", accuracy_score(y_test, predictions))
    print("Report:\n", classification_report(y_test, predictions))

    # Save model
    joblib.dump(model, 'disease_model.pkl')

# Step 3: Predict on new data
def predict_disease(new_data):
    model = joblib.load('disease_model.pkl')
    prediction = model.predict([new_data])
    return 'Disease detected' if prediction[0] == 1 else 'No disease'

if __name__ == "__main__":
    df = load_data()
train_model(df)
# Example patient data: [age, gender, fever, cough, bp_high, diabetes]
new_patient = [40, 1, 1, 1, 1, 0]
result = predict_disease(new_patient)
print("Prediction for new patient:", result)

if __name__ == "__main__":
    df = load_data()
train_model(df)
# Example patient data: [age, gender, fever, cough, bp_high, diabetes]
new_patient = [40, 1, 1, 1, 1, 0]
result = predict_disease(new_patient)
print("Prediction for new patient:", result)
```