

```
In [2]: # Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
In [4]: # Load your dataset (replace 'your_data.csv' with your dataset file)
data = pd.read_csv('heart_failure_clinical_records_dataset.csv')
```

```
In [5]: # Define features and target variable
X = data.drop('DEATH_EVENT', axis=1) # Features (input variables)
y = data['DEATH_EVENT'] # Target variable (0 or 1)
```

```
In [6]: # Split the dataset into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [7]: # Preprocess data (scaling)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
In [8]: # Initialize and train the model (Random Forest Classifier)
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

```
Out[8]: ▼      RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
In [9]: # Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.72	0.94	0.81	35
1	0.86	0.48	0.62	25
accuracy			0.75	60
macro avg	0.79	0.71	0.72	60
weighted avg	0.78	0.75	0.73	60

```
In [15]: # Load your training dataset (replace 'training_data.csv' with your dataset file)
training_data = pd.read_csv('heart_failure_clinical_records_dataset.csv')

# Define features and target variable for training data
X_train = training_data.drop('DEATH_EVENT', axis=1)
y_train = training_data['DEATH_EVENT']
```

```
In [16]: # Preprocess training data (scaling)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)

# Initialize and train the model (Random Forest Classifier)
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

```
Out[16]: ▼ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
In [22]: # Load the new patient data from an Excel file (replace 'new_patient_data.xlsx' with the file path)
new_patient_data = pd.read_excel('newdata.xlsx')
```

```
In [23]: # Preprocess the new patient data (scaling, same as the training data)
new_patient_data = scaler.transform(new_patient_data)
```

```
In [24]: # Make a prediction for the new patient
prediction = model.predict(new_patient_data)

# Interpret the prediction: 1 for heart failure predicted, 0 for no heart failure predicted
if prediction[0] == 1:
    print("Heart failure predicted for the new patient.")
```

```
else:  
    print("No heart failure predicted for the new patient.")
```

Heart failure predicted for the new patient.

In [ ]: