```
import pandas as pd
import numpy as np

Load the dataset #
df = pd.read_csv('C:/Users/lenovo/Desktop/datasets/heart_attack_Dataset1.csv')

Standardize categorical variables #
Gender #
})df['Gender'] = df['Gender'].str.lower().replace
,'m': 'male', 'f': 'female', 'ff': 'female', 'mm': 'male'
'other': 'other', ' ': 'unknown'
('fillna('unknown.({

SES #
})df['SES'] = df['SES'].str.lower().replace
'mid': 'middle', 'lo': 'low', ' ': 'unknown'
()fillna('unknown').str.capitalize.({

Smoking Status #
})df['Smoking Status'] = df['Smoking Status'].str.lower().replace
,'neverr': 'never', 'occasion': 'occasionally', 'nver': 'never'
'regular': 'regularly', ' ': 'unknown'
()fillna('unknown').str.capitalize.({

Binary variables (Hypertension, Diabetes, Family History) #
binary_cols = ['Hypertension', 'Diabetes', 'Family History of Heart Disease']
:for col in binary_cols
) = df[col]
df[col]
astype(str).
()str.strip().str.lower.
```

```
                replace({'yes': 1, 'no': 0, '1': 1, '0': 0, ' ': 0, 'nan': 0}).
                                                                                (
df[col] = pd.to_numeric(df[col], errors='coerce').fillna(0).astype(int)



                                                                Stress Level #
})df['Stress Level'] = df['Stress Level'].str.lower().replace
                                                  'med': 'medium', 'hi': 'high', 'lo': 'low', ' ': 'unknown'
                                                                ()fillna('unknown').str.capitalize.({


                                                                ECG Results #
})df['ECG Results'] = df['ECG Results'].str.lower().replace
                                                  ,'abnormal': 'abnormal', 'normal': 'normal', 'noormal': 'abnormal'
                                                                'unknown' :' '
                                                                ()fillna('unknown').str.capitalize.({


                                                                Handle numeric columns #
                                                  (100-18) Age - impute missing with median, cap reasonable range #
                                                  df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
                                                  (100 ,18)df['Age'] = df['Age'].fillna(df['Age'].median()).clip


                                                  Sleep Duration - reasonable range (3-12 hours) #
df['Sleep Duration (hrs/day)'] = pd.to_numeric(df['Sleep Duration (hrs/day)'],
                                                                                errors='coerce')
)df['Sleep Duration (hrs/day)'] = df['Sleep Duration (hrs/day)'].fillna
                                                  (df['Sleep Duration (hrs/day)'].median()).clip(3, 12


                                                  (300-100) Cholesterol - reasonable range #
df['Cholesterol Levels (mg/dL)'] = pd.to_numeric(df['Cholesterol Levels (mg/dL)'],
                                                                                errors='coerce')
)df['Cholesterol Levels (mg/dL)'] = df['Cholesterol Levels (mg/dL)'].fillna
                                                  (df['Cholesterol Levels (mg/dL)'].median()).clip(100, 300
```

```
BMI - fix extreme values (10-50 range) #
df['BMI (kg/m²)'] = pd.to_numeric(df['BMI (kg/m²)'], errors='coerce')
)df['BMI (kg/m²)'] = df['BMI (kg/m²)'].fillna
(df['BMI (kg/m²)'].median()).clip(10, 50

(220-60) Maximum Heart Rate - reasonable range #
df['Maximum Heart Rate Achieved'] = pd.to_numeric(df['Maximum Heart Rate Achieved'],
errors='coerce')
)df['Maximum Heart Rate Achieved'] = df['Maximum Heart Rate Achieved'].fillna
(df['Maximum Heart Rate Achieved'].median()).clip(60, 220

(100-90) Blood Oxygen Levels - reasonable range #
df['Blood Oxygen Levels (SpO2%)'] = pd.to_numeric(df['Blood Oxygen Levels (SpO2%)'],
errors='coerce')
)df['Blood Oxygen Levels (SpO2%)'] = df['Blood Oxygen Levels (SpO2%)'].fillna
(df['Blood Oxygen Levels (SpO2%)'].median()).clip(90, 100

Heart Attack Likelihood - convert to binary #
})df['Heart Attack Likelihood'] = df['Heart Attack Likelihood'].astype(str).str.lower().replace
,yes': 1'
,no': 0'
,noo': 0'
,0 :' '
nan': 0'
(fillna(0).astype(int.({

One-hot encoding for categorical variables #
categorical_cols = ['Gender', 'SES', 'Smoking Status', 'Stress Level', 'ECG Results']
df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
```

```
Final check for any remaining missing values #
df = df.dropna()  # or use more sophisticated imputation if needed


Save cleaned data #
df.to_csv('cleaned_heart_attack_data.csv', index=False)
```