In [243…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, StandardScaler
import joblib
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

In [244…
```
df = pd.read_csv('heart_attack_prediction_indonesia.csv')[:30000]
clean_df = pd.read_csv('clean_hap.csv')
# df
```

This report presents a predictive health data analysis aimed at identifying factors contributing to heart attacks based on clinical, lifestyle, and demographic variables. The dataset comprises patient health profiles collected through medical screenings and surveys.

In [245…
```
df.shape
```

Out[245… (30000, 28)

Data Description (before cleaning and preprocessing)

Data Summary

- Total Records: 30k rows
- Total Features: 34 (including the target variable heart_attack)

age, gender, region, income_level, hypertension, diabetes, cholesterol_level, obesity, waist_circumference, family_history, smoking_status, alcohol_consumption, physical_activity, dietary_habits, air_pollution_exposure, stress_level, sleep_hours, blood_pressure_systolic, blood_pressure_diastolic, fasting_blood_sugar, cholesterol_hdl, cholesterol_ldl, triglycerides, EKG_results, previous_heart_disease, medication_usage, participated_in_free_screening, heart_attack, age_group

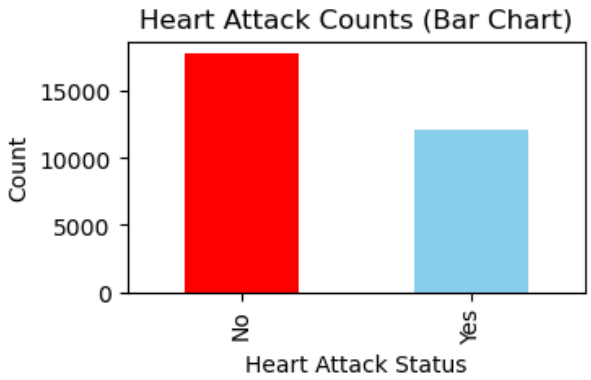
Distribution of heart attack feature

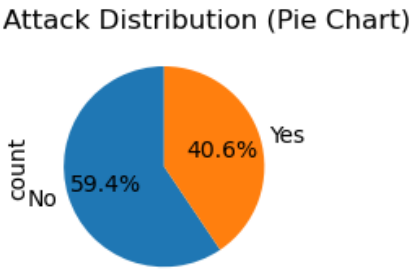
The target variable heart_attack is binary:

- 0: No heart attack
- 1: Experienced a heart attack

In [246…
```
counts = df.heart_attack.value_counts()
plt.figure(figsize=(8,2))
plt.subplot(1,2,1)
counts.plot(kind='bar', color=['red','skyblue'], legend=False)
plt.xlabel('Heart Attack Status')
```

```
plt.ylabel('Count')
plt.xticks([0,1],['No','Yes'])
plt.title('Heart Attack Counts (Bar Chart)')
plt.subplot(1,2,2)
counts.plot(kind='pie', autopct='%1.1f%%', labels=['No','Yes'], color=['red','sk
plt.title('Heart Attack Distribution (Pie Chart)')
plt.show()
```





🧬 Types of Features

The dataset includes a comprehensive set of demographic, clinical, and lifestyle variables, enabling robust analysis of cardiovascular risk.

📊 Demographic Features

- **age**: Integer (Range: 25–90)
- **gender**: Categorical (`'Male'` , `'Female'`)
- **region**: Categorical (`'Urban'` , `'Rural'`)
- **income_level**: Categorical (`'Low'` , `'Middle'` , `'High'`)

```
#### 🩺 Clinical Features

- **hypertension**: Binary (0 = No, 1 = Yes)
- **diabetes**: Binary (0 = No, 1 = Yes)
- **cholesterol_level**: Numeric
- **blood_pressure_systolic**: Numeric
- **blood_pressure_diastolic**: Numeric
- **fasting_blood_sugar**: Numeric
- **cholesterol_hdl**: Numeric
- **cholesterol_ldl**: Numeric
- **triglycerides**: Numeric
- **obesity**: Binary (0 = No, 1 = Yes)
- **waist_circumference**: Numeric
- **previous_heart_disease**: Binary (0 = No, 1 = Yes)
- **medication_usage**: Binary (0 = No, 1 = Yes)
- **EKG_results**: Categorical (`'Normal'`, `'Abnormal'`)
```

💡 Lifestyle Features

- **smoking_status**: Categorical (`'Never'`, `'Past'`, `'Current'`)
- **alcohol_consumption**: Categorical (`'Moderate'`, `'High'`, `nan`)
- **physical_activity**: Categorical (`'Low'`, `'Moderate'`, `'High'`)
- **dietary_habits**: Categorical (`'Healthy'`, `'Unhealthy'`)
- **air_pollution_exposure**: Categorical (`'Low'`, `'Moderate'`, `'High'`)
- **stress_level**: Categorical (`'Low'`, `'Moderate'`, `'High'`)
- **sleep_hours**: Continuous Numeric
- **family_history**: Binary (0 = No, 1 = Yes)
- **participated_in_free_screening**: Binary (0 = No, 1 = Yes)

📌 **Note**:
This dataset's richness across clinical and behavioral dimensions supports deep exploratory data analysis (EDA) and the development of predictive models for cardiovascular events like heart attacks.

In [247…

```
unique_vals = []
for i in df.columns:
    unique_vals.append({f'{i}:{df[i].unique()}'})
unique_vals
```

Out[247… [{'age:[60 53 62 73 52 64 49 61 57 32 34 48 42 58 44 38 72 55 37 56 41 59 47 51
\n 77 54 40 31 39 63 46 67 33 50 66 71 25 45 65 84 68 81 43 36 70 35 87 90\n 82
80 30 76 74 29 69 79 78 27 75 28 85 86 83 26 88 89]'},
{"gender:['Male' 'Female']"},
{"region:['Rural' 'Urban']"},
{"income_level:['Middle' 'Low' 'High']"},
{'hypertension:[0 1]'},
{'diabetes:[1 0]'},
{'cholesterol_level:[211 208 231 202 232 238 165 186 121 196 190 234 193 125 1
34 271 185 230\n 132 163 200 191 219 142 180 205 228 265 177 192 176 207 174 22
5 170 130\n 251 201 159 172 153 258 221 189 214 105 255 149 128 199 131 139 133
168\n 188 285 252 216 220 212 116 182 250 175 246 215 226 240 210 147 254 227\n
243 223 146 241 173 256 244 198 247 187 164 217 218 152 161 303 203 206\n 245 1
51 249 181 178 183 166 184 162 204 156 179 171 158 270 222 136 194\n 154 253 16
7 236 209 266 263 148 242 195 239 235 118 113 229 123 117 197\n 233 273 224 272
150 310 298 169 257 268 155 145 213 248 237 160 141 259\n 111 275 144 127 264 1
38 277 305 157 280 119 103 260 135 261 279 140 137\n 274 114 115 122 107 100 27
8 124 283 318 143 306 288 287 120 276 299 269\n 129 325 262 106 319 112 293 284
301 281 295 302 296 292 126 289 108 282\n 109 267 102 290 300 286 308 312 326 3
09 291 104 311 330 316 334 294 304\n 315 317 341 110 335 323 307 297 324 321 10
1 314 313 336 320 331 339 350\n 327 349 337 328 329]'},
{'obesity:[0 1]'},
{'waist_circumference:[83 106 112 82 89 81 91 72 115 88 99 80 70 74
77 63 116 95\n 101 84 73 113 132 104 100 90 117 110 92 94 105 97 76 9
8 108 122\n 93 86 60 124 109 75 148 66 67 111 87 71 96 107 114 69 61
102\n 85 62 103 68 78 58 79 65 130 119 121 123 129 120 135 59 57 53\n
44 50 127 118 131 126 54 143 55 134 128 64 56 133 139 43 52 51\n 125 4
9 46 48 136 45 42 138 141 140 137 173 144 142 151 47 146 34\n 27 149 145
37 155 40 41 33 147 152 150 31 154 20]'},
{'family_history:[0 1]'},
{"smoking_status:['Never' 'Past' 'Current']"},
{"alcohol_consumption:[nan 'Moderate' 'High']"},
{"physical_activity:['High' 'Moderate' 'Low']"},
{"dietary_habits:['Unhealthy' 'Healthy']"},
{"air_pollution_exposure:['Moderate' 'High' 'Low']"},
{"stress_level:['Moderate' 'High' 'Low']"},
{'sleep_hours:[5.97060316 5.64381314 6.33619667 ... 6.71533855 6.44848813 5.89
14506]'},
{'blood_pressure_systolic:[113 132 116 136 127 131 128 109 150 142 98 143 146
145 110 138 129 99\n 122 123 117 137 121 130 106 141 118 90 85 100 162 104 1
26 144 166 125\n 114 135 140 148 107 139 155 151 149 124 115 103 105 119 152 10
2 147 161\n 134 112 78 153 169 133 120 111 101 159 164 83 88 108 92 94 82
157\n 170 97 96 154 163 165 91 89 160 95 156 158 93 168 187 81 167 173\n
74 171 72 174 180 87 86 175 176 172 77 79 84 80 182 181 73 179\n 76 17
8 75 177 190 183 71]'},
{'blood_pressure_diastolic:[62 76 74 65 75 71 97 83 87 98 103 77 5
4 64 81 49 79 93\n 67 70 89 66 85 78 80 86 105 82 72 68 73 92
90 96 88 61\n 84 69 91 95 94 63 102 101 59 99 60 58 108 106 107 5
6 100 51\n 104 55 53 57 52 109 113 111 48 45 46 50 114 110 40 43 112
47\n 115 116 44 127]'},
{'fasting_blood_sugar:[173 70 118 98 104 129 88 112 147 79 105 113 126 86
83 84 141 145\n 101 114 102 157 127 117 107 89 109 153 108 80 140 168 73 7
5 94 123\n 77 144 119 81 92 116 125 100 135 71 165 85 121 111 120 90 137
164\n 156 131 103 134 161 158 110 76 74 142 115 143 146 91 162 148 154 93\n
167 132 130 72 160 96 124 97 122 95 133 170 151 150 149 136 176 99\n 182
78 82 187 178 106 172 155 152 138 186 207 177 128 169 139 87 174\n 166 195 18
5 171 159 163 181 179 184 210 203 220 180 175 194 196 216 212\n 183 190 188 202
201 192 193 199 197 215 198 191 189 200 204 206 214 208\n 209]'},
{'cholesterol_hdl:[48 58 69 52 59 34 40 47 46 38 62 39 44 61 55 50 56 35 49 45
66 51 53 33\n 37 57 64 54 63 43 32 60 19 72 27 41 30 67 28 36 31 23 42 24 68 73

65 76\n 20 74 70 26 21 71 25 29 81 77 75 18 78 79 17 22 15 83 80 16 85 87 86 82
\n 13 88 89 14]'},
 {'cholesterol_ldl:[121 83 130 85 127 148 128 100 157 93 109 172 89 60 185
175 126 158\n 144 203 205 141 132 72 116 110 53 140 161 107 33 165 159 97 1
77 152\n 195 117 99 82 193 171 108 137 162 119 150 215 153 134 84 131 135 14
2\n 123 90 155 71 115 77 105 106 166 133 145 101 118 149 111 183 187 95\n
98 103 180 163 173 129 156 167 57 188 112 113 189 169 88 124 114 74\n 219 13
8 160 104 70 164 76 184 122 186 170 81 182 226 143 146 210 154\n 66 62 102
96 87 125 181 197 80 139 51 147 120 204 174 42 168 178\n 136 79 67 199 7
3 179 86 208 194 92 65 94 91 176 235 64 78 207\n 191 52 196 61 45 198
151 240 55 59 58 221 54 39 201 46 200 192\n 69 206 21 209 63 47 49
34 68 222 75 202 216 24 214 30 213 56\n 233 38 190 43 50 211 35 36 27
5 37 48 31 20 40 230 232 241 238\n 223 225 16 252 218 28 244 256 15 237
231 217 212 262 44 32 220 26\n 261 27 17 228 29 224 41 243 22 229 227
9 258 11 260 239 19 -13\n 234 12   3 246 -7 23   8 253   1 13 236 250 247
242]'},
 {'triglycerides:[101 138 171 146 139 191 167 50 198 164 145 148 92 186 202 1
76 157 233\n 155 170 183 188 211 152 185 154 147 247 161 192 133 205 112 78 14
1 212\n 162 196 184 110 125 163 76 230 215 74 119 187 168 190 178 89 87 156
\n 201 159 172 200 140 84 102 126 239 85 197 222 169 223 107 275 254 165\n 8
0 179 111 244 127 144 130 181 182 153 136 241 79 149 137 124 108 113\n 150 86
189 135 105 72 73 194 174 142 70 242 121 166 177 214 118 206\n 158 240 97 1
22 238 57 160 98 100 131 128 60 123 199 104 106 83 143\n 300 94 151 116 8
8 204 226 117 234 209 180 59 245 268 216 71 77 64\n 224 250 208 91 132 93
263 75 262 227 95 173 203 236 269 67 134 99\n 258 252 175 96 129 115 253 1
14 120 249 259 193 195 109 219 213 103 62\n 225 81 243 282 237 228 217 54 5
8 264 69 207 229 218 66 246 61 232\n 220 248 55 56 274 221 278 235 65 90
52 270 210 265 82 297 63 251\n 51 295 279 68 255 231 266 323 257 298 267 26
1 288 276 53 285 272 307\n 271 256 286 316 291 281 318 296 283 340 294 303 305
273 260 301 308 317\n 309 290 299 284 277 287 334 342 293 314 330 306 329 322 3
02 320 280 289\n 319 327 349]'},
 {"EKG_results:['Normal' 'Abnormal']"},
 {'previous_heart_disease:[0 1]'},
 {'medication_usage:[0 1]'},
 {'participated_in_free_screening:[0 1]'},
 {'heart_attack:[0 1]'}]

Exploratory Data Analysis

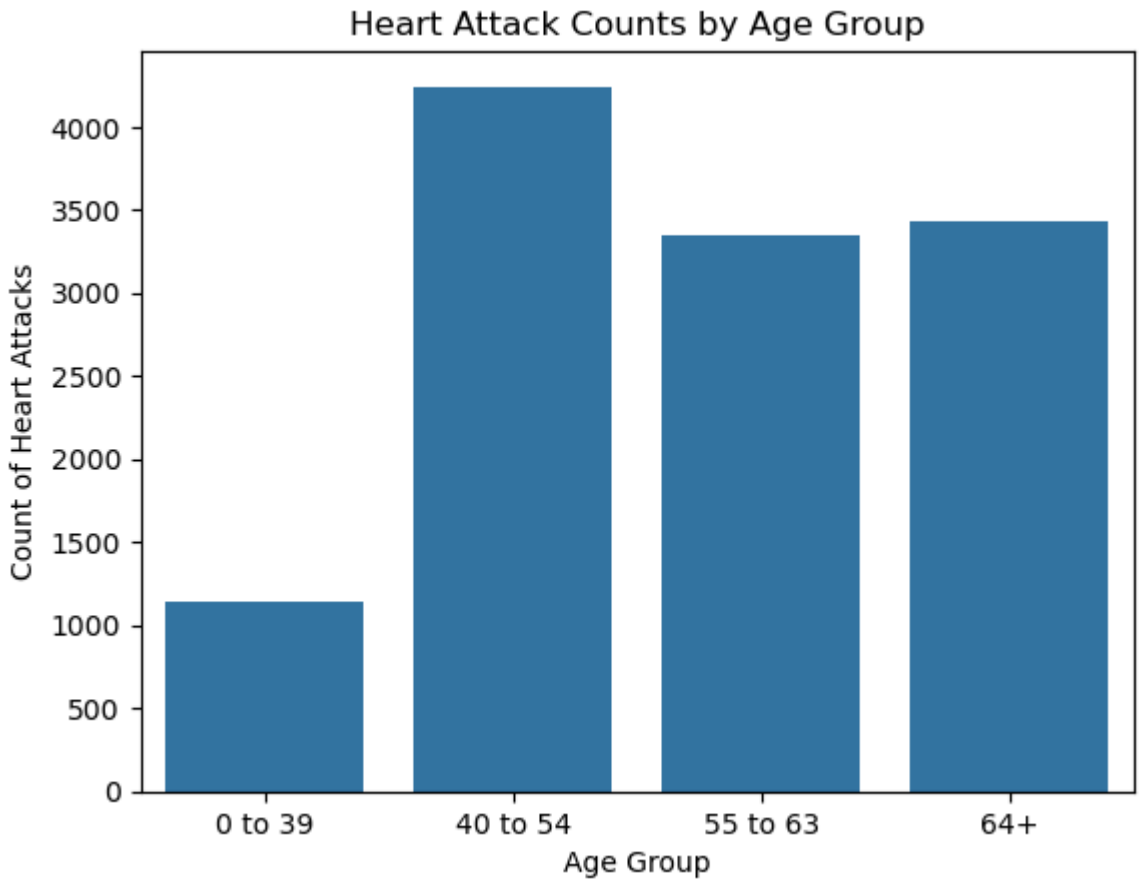
Focus: What features show meaningful differences between patients with and without heart attacks?

Numeric Features

In [248…

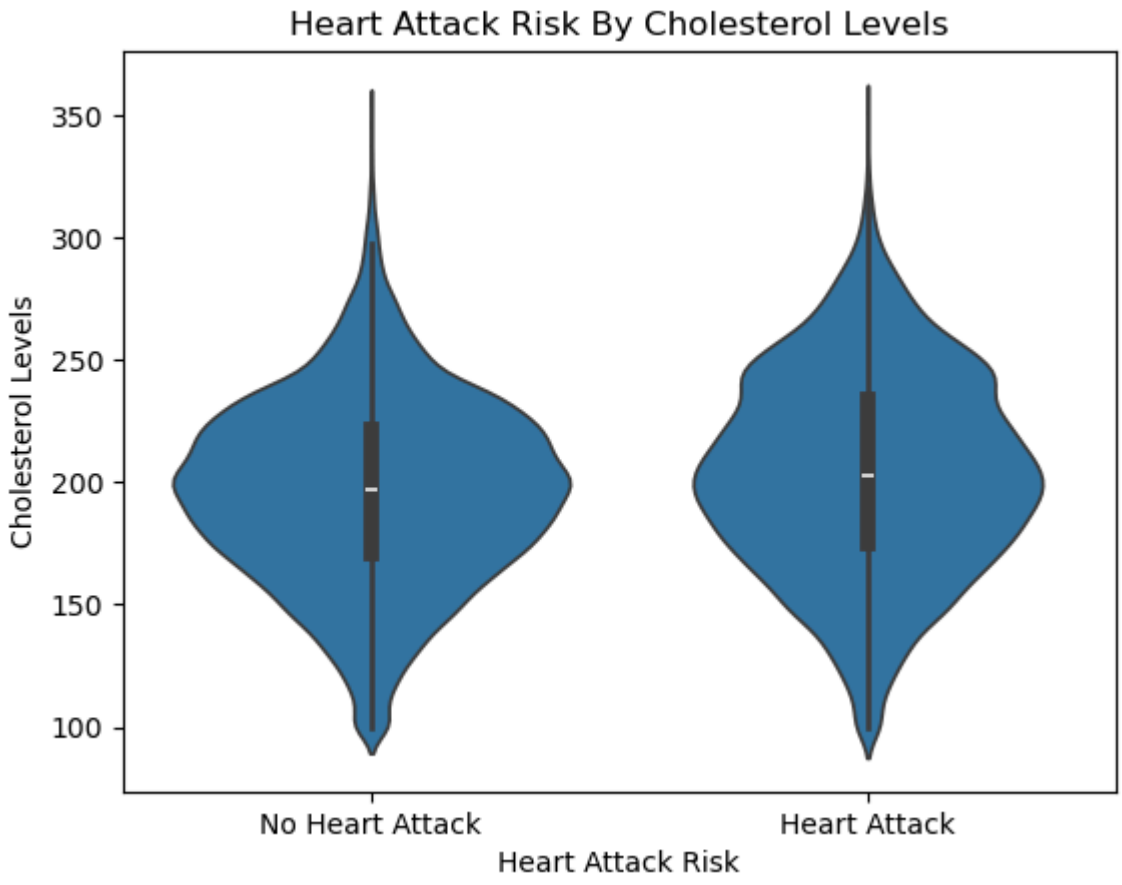
```
df['age_group'] = np.where(
    df['age'] < 40, 1, np.where(
        df['age'] < 55, 2, np.where(
            df['age'] < 64, 3, 4
        )
    )
)
counts = df.groupby('age_group')['heart_attack'].value_counts().unstack()
counts.columns = ['No Heart Attack', 'Heart Attack']
sns.barplot(data=counts.reset_index(), x='age_group', y='Heart Attack')
plt.xticks([0, 1, 2, 3], ['0 to 39', '40 to 54', '55 to 63', '64+'])
plt.xlabel('Age Group')
plt.ylabel('Count of Heart Attacks')
```

```
plt.title('Heart Attack Counts by Age Group')
plt.show()
```



In [249…

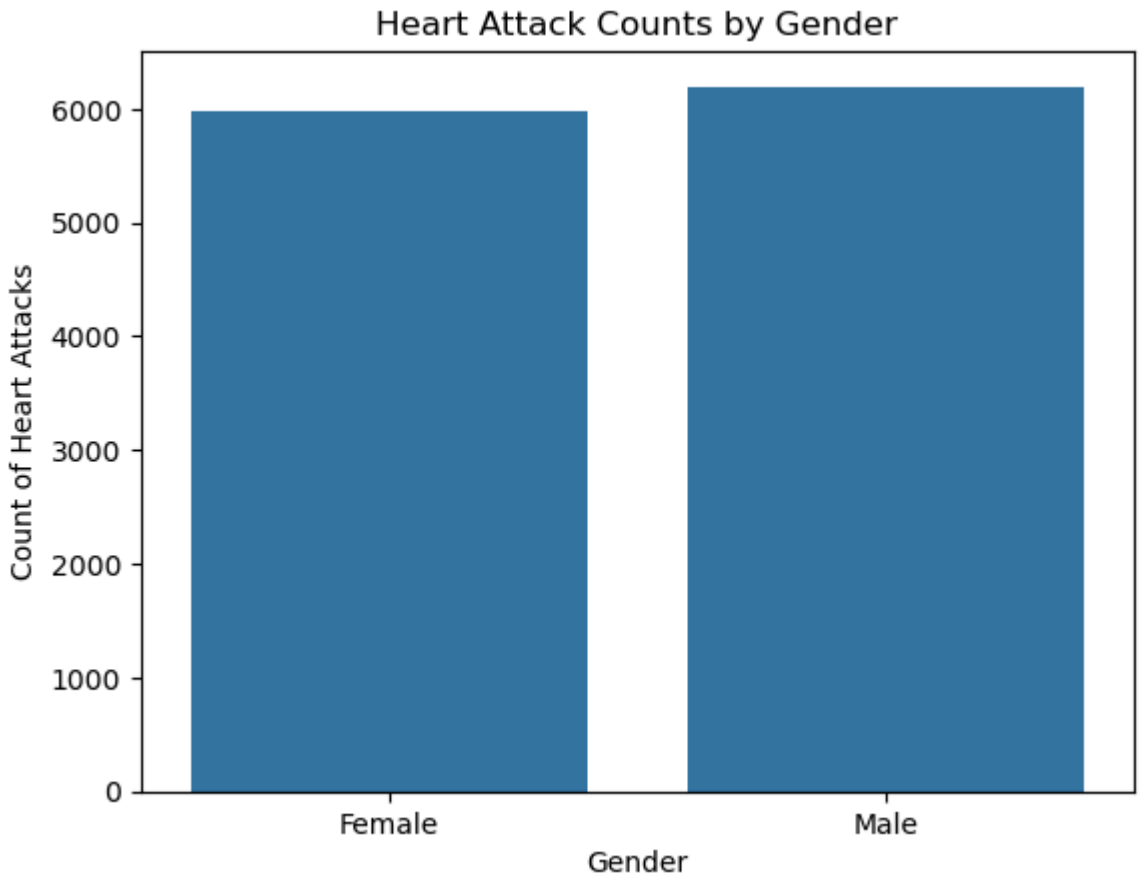
```
sns.violinplot(data=df, x='heart_attack', y='cholesterol_level')
plt.xticks([0,1],['No Heart Attack', 'Heart Attack'])
plt.title('Heart Attack Risk By Cholesterol Levels')
plt.xlabel('Heart Attack Risk')
plt.ylabel('Cholesterol Levels')
plt.show()
```



Categorical Features

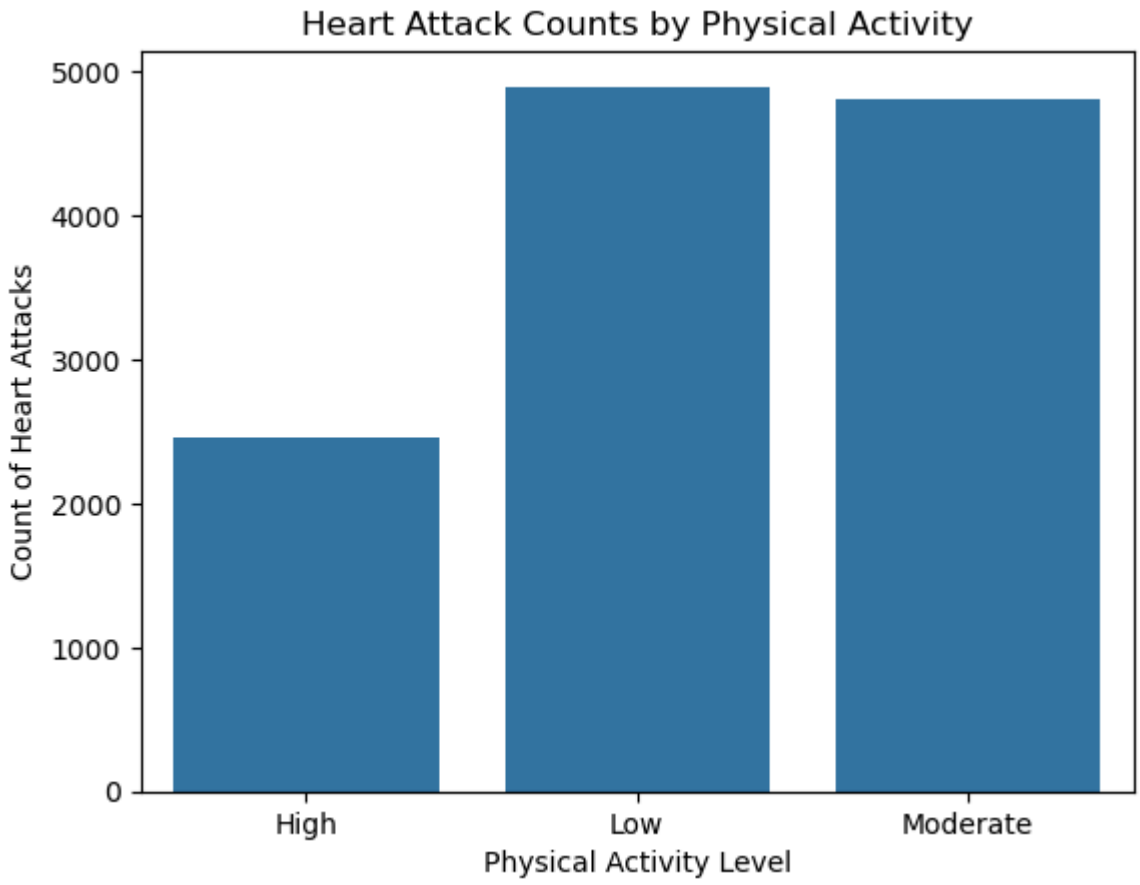
In [250…

```
counts = df.groupby('gender')['heart_attack'].value_counts().unstack()
counts.columns = ['No Heart Attack', 'Heart Attack']
sns.barplot(data=counts.reset_index(), x='gender', y='Heart Attack')
plt.xlabel('Gender')
plt.ylabel('Count of Heart Attacks')
plt.title('Heart Attack Counts by Gender')
plt.show()
```



In [251…

```
counts = df.groupby('physical_activity')['heart_attack'].value_counts().unstack(
counts.columns = ['No Heart Attack', 'Heart Attack']
sns.barplot(data=counts.reset_index(), x='physical_activity', y='Heart Attack')
plt.xlabel('Physical Activity Level')
plt.ylabel('Count of Heart Attacks')
plt.title('Heart Attack Counts by Physical Activity')
plt.show()
```



Correlations (numeric features only)

In [252…

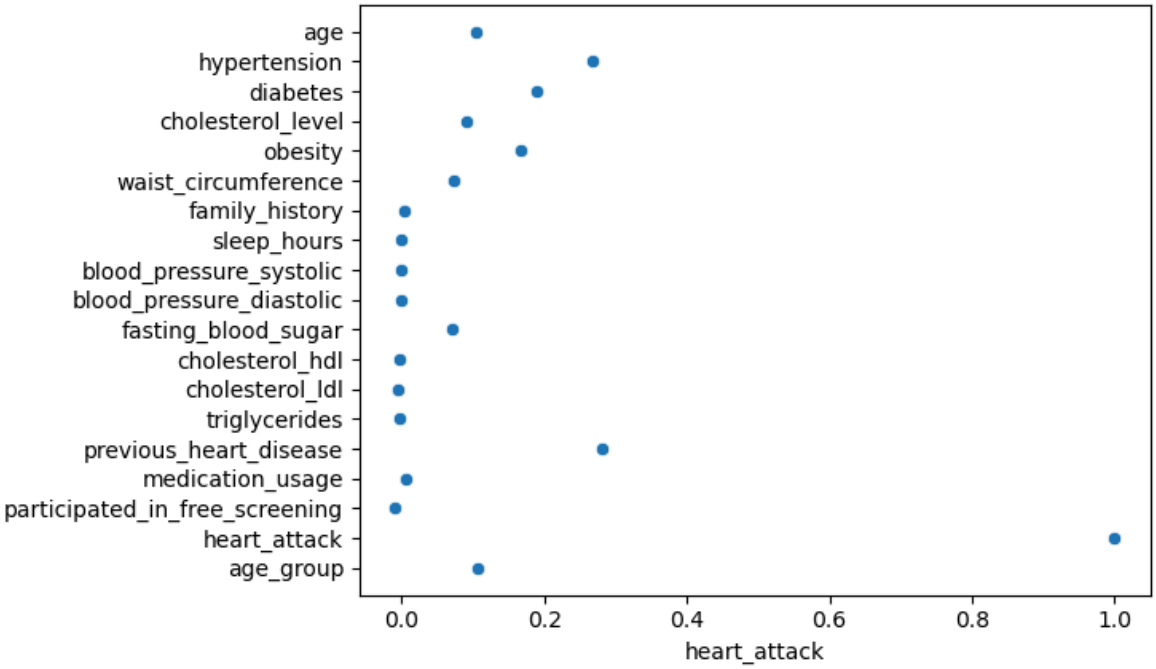
```
num_df = df.select_dtypes(include='number')
num_corrs = num_df.corr()[['heart_attack']].sort_values(by='heart_attack', ascen
num_corrs = num_corrs[1:]
num_corrs
```

Out[252...

	heart_attack
previous_heart_disease	0.281420
hypertension	0.267961
diabetes	0.189587
obesity	0.166805
age_group	0.107021
age	0.105092
cholesterol_level	0.091452
waist_circumference	0.073093
fasting_blood_sugar	0.071354
medication_usage	0.005347
family_history	0.002394
blood_pressure_diastolic	-0.000030
blood_pressure_systolic	-0.001201
sleep_hours	-0.001916
triglycerides	-0.002354
cholesterol_hdl	-0.003616
cholesterol_ldl	-0.004460
participated_in_free_screening	-0.009094

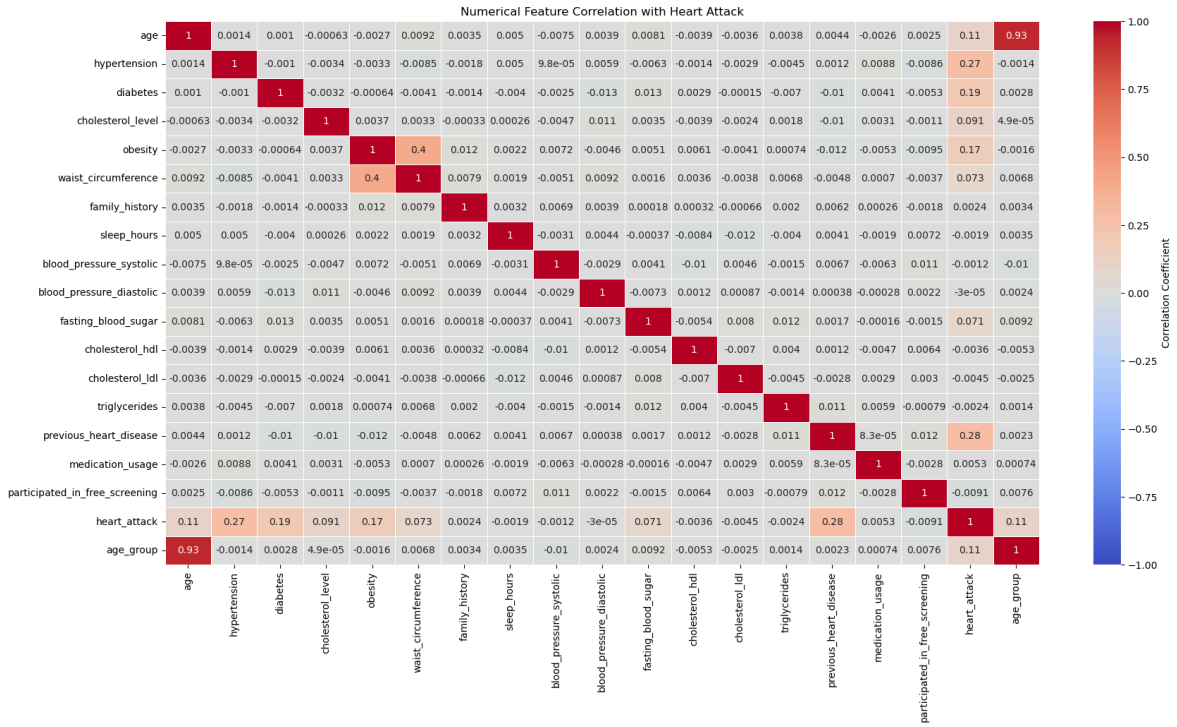
In [253...

```
sns.scatterplot(data=num_df.corr(), x='heart_attack', y=num_df.columns.tolist())
plt.show()
```



In [254...

```
plt.figure(figsize=(20,10))
sns.heatmap(num_df.corr(),
            annot=True,
            cmap='coolwarm',
            vmin=-1,
            vmax=1,
            linewidths=0.5,
            cbar_kws={'label': 'Correlation Coefficient'})
plt.title('Numerical Feature Correlation with Heart Attack')
plt.show()
```



Feature Engineering

Several new features were derived:

- age_group: Binned age into categories (e.g., 18–30, 31–45...)
- health_risk_score: Composite risk index based on conditions like hypertension, cholesterol, diabetes
- obesity_risk_score: Derived from waist_circumference, BMI (if available), and obesity status
- stress_to_sleep_ratio: stress_level / sleep_hours
- mean_arterial_pressure: Calculated as: MAP = Diastolic BP + 1/3 (Systolic BP − Diastolic BP)
- triglyceride-hdl-ratio: triglycerides / cholesterol_hdl

Encoding Categorical Variables:

Categorical variables such as gender, region, and EKG_results were encoded using label encoding:

- gender_encoded: 0 = Female, 1 = Male

- region_encoded: numeric mapping of regions
- alcohol_consumption: ordinal encoding reflecting severity

In [255…

```
df['alcohol_consumption'] = df['alcohol_consumption'].fillna('None')
stress_mapping = {
    'Low': 1,
    'Moderate': 2,
    'High': 3
}
df['stress_level'] = df['stress_level'].map(stress_mapping)
smoking_mapping = {
    'Never': 1,
    'Past': 2,
    'Current': 3
}
df['smoking_status'] = df['smoking_status'].map(smoking_mapping)
alcohol_consumption_mapping = {
    'None': 1,
    'Moderate': 2,
    'High': 3
}
df['alcohol_consumption'] = df['alcohol_consumption'].map(alcohol_consumption_ma
physical_activity_mapping = {
    'Low': 1,
    'Moderate': 2,
    'High': 3
}
df['physical_activity'] = df['physical_activity'].map(physical_activity_mapping)
dietary_habits_mapping = {
    'Unhealthy': 1,
    'Healthy': 2,
}
df['dietary_habits'] = df['dietary_habits'].map(dietary_habits_mapping)
air_pollution_mapping = {
    'Low': 1,
    'Moderate': 2,
    'High': 3
}
df['air_pollution_exposure'] = df['air_pollution_exposure'].map(air_pollution_ma
income_level_mapping = {
    'Low': 3,
    'Middle': 2,
    'High': 1
}
df['income_level'] = df['income_level'].map(income_level_mapping)
df['age_group'] = np.where(
    df['age'] < 40, 1, np.where(
        df['age'] < 55, 2, np.where(
            df['age'] < 64, 3, np.where(
                df['age'] < 65, 4, 5
            )
        )
    )
)
df['health_risk_score'] = (
    df['hypertension'] +
    df['diabetes'] +
    df['obesity'] +
    df['family_history'] +
```

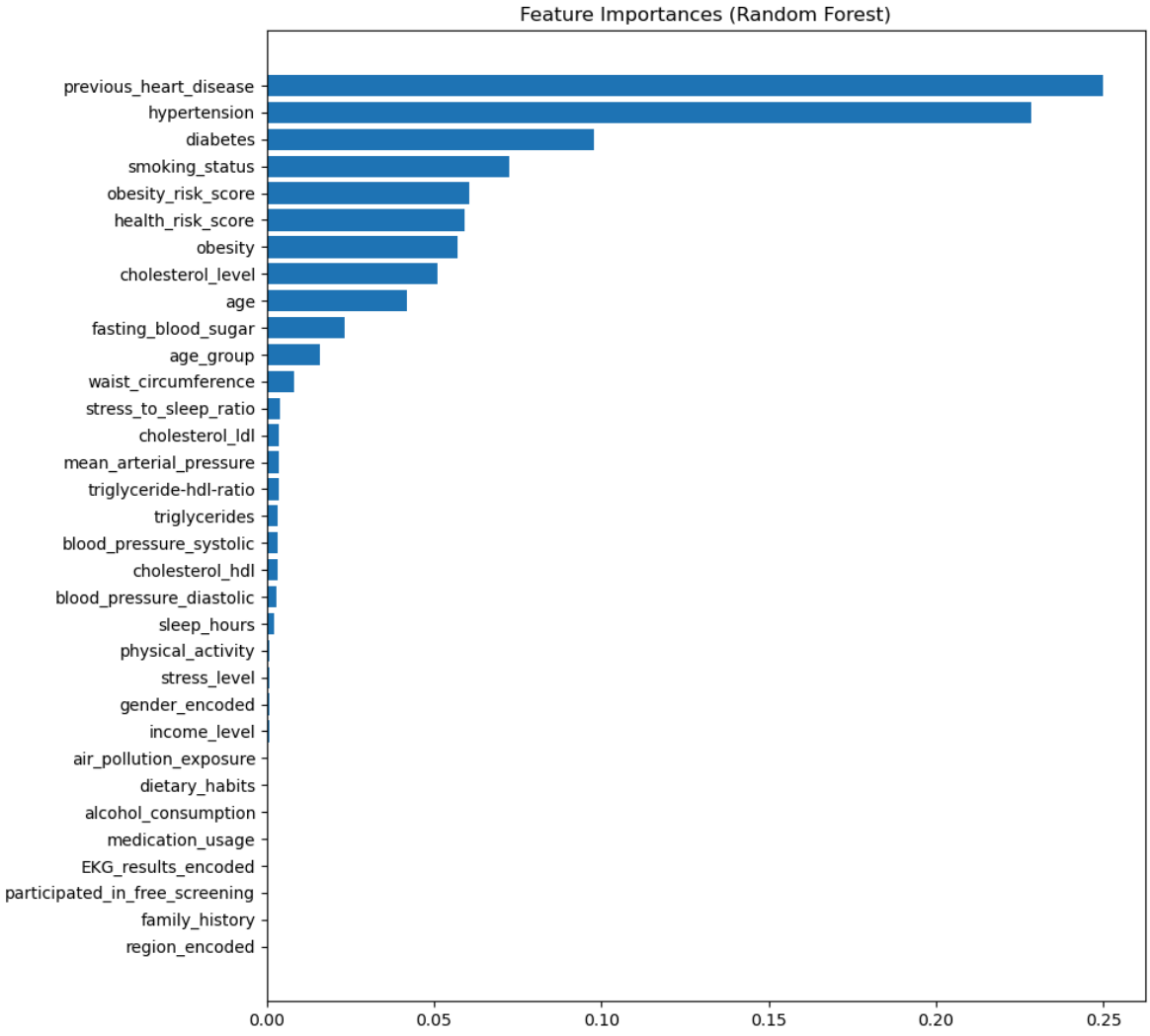
```
    df['smoking_status'] +
    df['alcohol_consumption'] +
    (1 - df['physical_activity']) +
    (1 - df['dietary_habits']) +
    df['air_pollution_exposure'] +
    df['stress_level'] +
    df.income_level
)
df['obesity_risk_score'] = np.where(df['gender'] == 'Male' , df['obesity'] * 1 +
df['stress_to_sleep_ratio'] = df['stress_level'] / df['sleep_hours']
df['mean_arterial_pressure'] = (2* df['blood_pressure_systolic'] + df['blood_pre
df['triglyceride-hdl-ratio'] = df['triglycerides'] / df['cholesterol_hdl']
cat_df = df.select_dtypes(include='object')
num_df = df.select_dtypes(exclude='object')
encoder = LabelEncoder()
for cols in cat_df:
    cat_df[cols+'_encoded'] = encoder.fit_transform(cat_df[cols])
cat_df = cat_df.select_dtypes(exclude='object')
df = pd.concat([cat_df, num_df], axis=1)
```

Feature Importances & Modelling

Focus: What features actually help predict heart attacks?

In [256…

```
x = clean_df.drop('heart_attack', axis=1)
y = clean_df['heart_attack']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_
SEED = 42
rf =  RandomForestClassifier(
    criterion='entropy',
    max_depth=5,
    max_features='sqrt',
    min_samples_leaf=1,
    min_samples_split=3,
)
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
rf.fit(X_train_scaled, y_train)
importances = rf.feature_importances_
feature_names = X_train.columns
importance_df = pd.DataFrame({'Feature': feature_names, 'Importance': importance
importance_df.sort_values(by='Importance', ascending=False, inplace=True)
plt.figure(figsize=(10,9))
plt.barh(importance_df['Feature'], importance_df['Importance'])
plt.title("Feature Importances (Random Forest)")
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```



📝 5. Insights & Recommendations

🔍 Key Risk Indicators (Top Features)

Based on the Random Forest feature importance analysis, the top predictors of heart attack risk are:

1. **Previous heart disease**
2. **Hypertension**
3. **Diabetes**
4. **Smoking status**
5. **Obesity**

These features show the strongest associations with heart attack outcomes and should be prioritized in risk assessments and screening strategies.

🌏 Public Health Impact in Indonesia

- **Who is Most at Risk**
  Individuals with a history of heart disease, hypertension, or diabetes, as well as

current smokers and those who are obese, face the highest risk. This is especially concerning among aging populations in both urban and rural settings.

- **How Early Can We Detect?**
  Many of the top risk factors—such as blood pressure, blood sugar levels, and body mass index—can be measured early and relatively easily, enabling timely intervention.

- **Feasibility in Rural Clinics**
  Features like blood pressure, obesity (via BMI), and smoking status require minimal equipment and can be assessed by trained health workers, making them suitable for use in rural or under-resourced areas.

Policy & Clinical Use

- **Integration into Screening Programs**
  The top features identified can be used to design or enhance screening tools for early detection. Community-based programs and free health screenings could focus on these specific indicators to identify high-risk individuals.

- **Mobile App Potential**
  A small set of these high-impact features could power a simple risk calculator or mobile health app, aiding both individuals and community health workers in identifying those at risk.

- **Promoting Affordable, Data-Based Prevention**
  Interventions targeting modifiable factors like smoking, hypertension, and obesity offer a cost-effective approach for reducing cardiovascular risk in the population, especially in lower-income and rural communities.