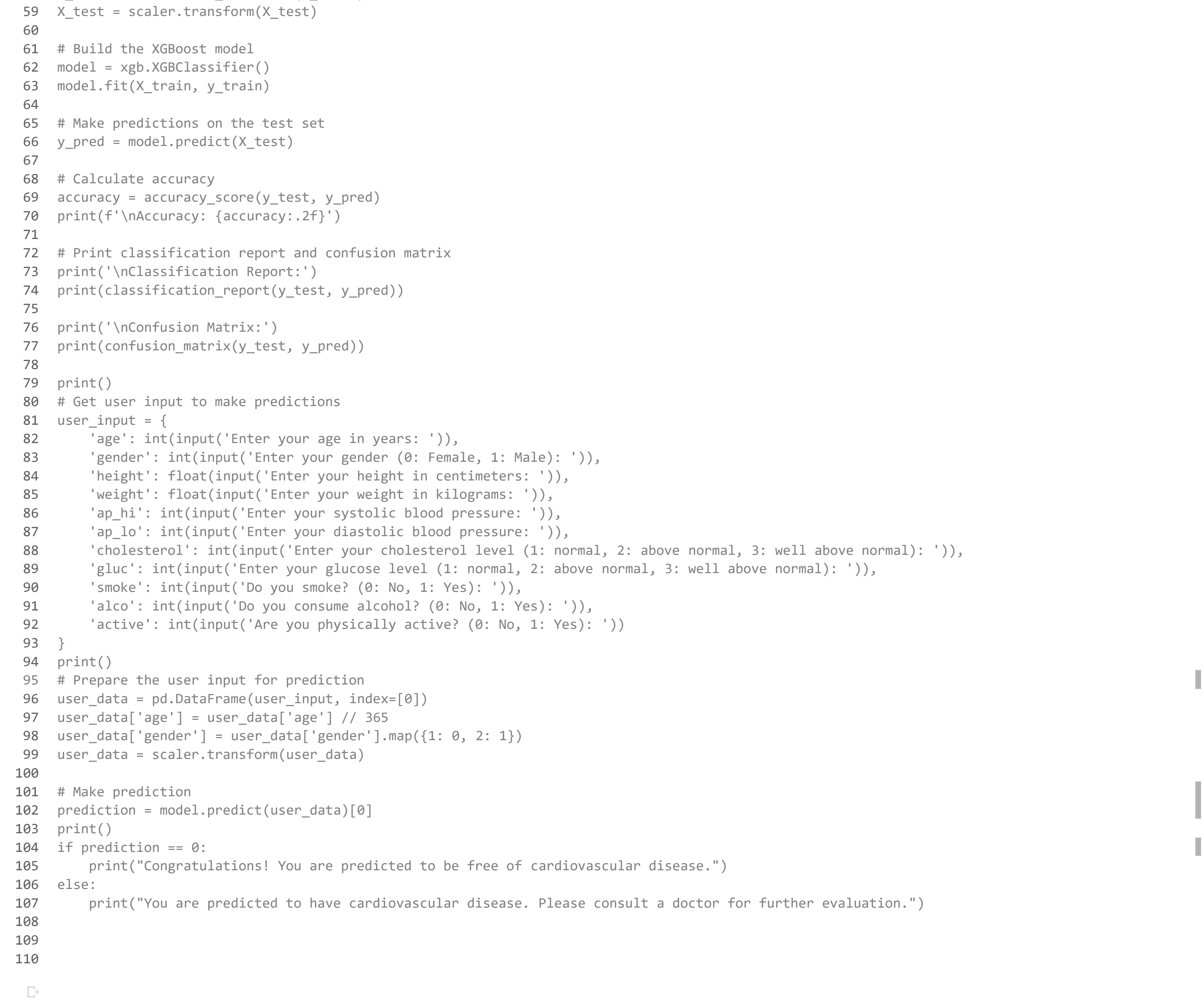
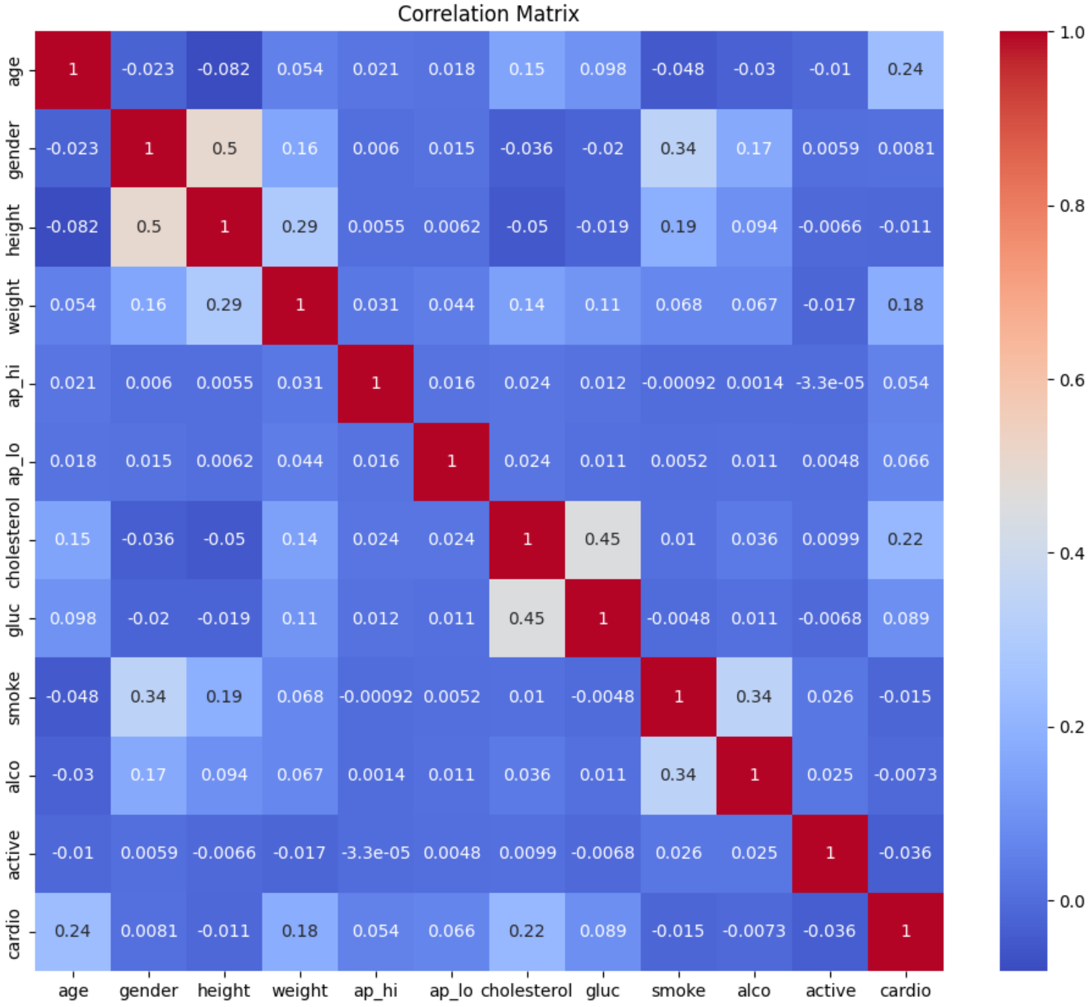
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import xgboost as xgb

# Load the CSV data
data = pd.read_csv('/content/sample_data/cardio_dataset.csv')

# Drop the 'id' column as it is not required for modeling
data.drop(columns=['id'], inplace=True)

# Convert age from days to years
data['age'] = data['age'] // 365

# Convert gender to categorical (0: Female, 1: Male)
data['gender'] = data['gender'].map({1: 0, 2: 1})

# Check for and handle any missing values (if present)
data.dropna(inplace=True)

# Split the data into features (X) and target (y)
X = data.drop(columns=['cardio'])
y = data['cardio']

# Show basic statistics of the dataset
print(data.describe())
print()

# Show correlation matrix
correlation_matrix = data.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.show()
print()

# Plot distributions of numerical features
num_features = ['age', 'height', 'weight', 'ap_hi', 'ap_lo']
for feature in num_features:
    sns.histplot(data[feature], kde=True)
    plt.title(f'Distribution of {feature}')
    plt.show()
print()
# Plot count of each category for categorical features
cat_features = ['cholesterol', 'gluc', 'smoke', 'alco', 'active']
for feature in cat_features:
    sns.countplot(data[feature])
    plt.title(f'Count of {feature}')
    plt.show()
print()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
```



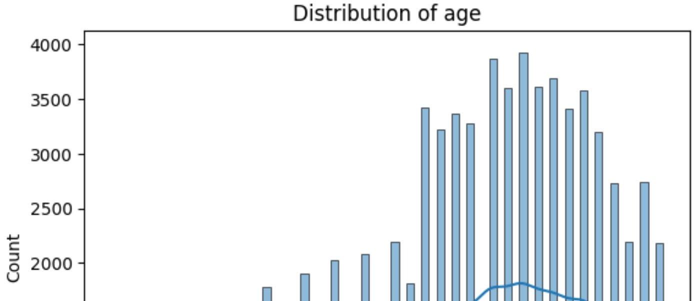
```python
X_test = scaler.transform(X_test)

# Build the XGBoost model
model = xgb.XGBClassifier()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'\nAccuracy: {accuracy:.2f}')

# Print classification report and confusion matrix
print('\nClassification Report:')
print(classification_report(y_test, y_pred))

print('\nConfusion Matrix:')
print(confusion_matrix(y_test, y_pred))

print()
# Get user input to make predictions
user_input = {
    'age': int(input('Enter your age in years: ')),
    'gender': int(input('Enter your gender (0: Female, 1: Male): ')),
    'height': float(input('Enter your height in centimeters: ')),
    'weight': float(input('Enter your weight in kilograms: ')),
    'ap_hi': int(input('Enter your systolic blood pressure: ')),
    'ap_lo': int(input('Enter your diastolic blood pressure: ')),
    'cholesterol': int(input('Enter your cholesterol level (1: normal, 2: above normal, 3: well above normal): ')),
    'gluc': int(input('Enter your glucose level (1: normal, 2: above normal, 3: well above normal): ')),
    'smoke': int(input('Do you smoke? (0: No, 1: Yes): ')),
    'alco': int(input('Do you consume alcohol? (0: No, 1: Yes): ')),
    'active': int(input('Are you physically active? (0: No, 1: Yes): '))
}
print()
# Prepare the user input for prediction
user_data = pd.DataFrame(user_input, index=[0])
user_data['age'] = user_data['age'] // 365
user_data['gender'] = user_data['gender'].map({1: 0, 2: 1})
user_data = scaler.transform(user_data)

# Make prediction
prediction = model.predict(user_data)[0]
print()
if prediction == 0:
    print("Congratulations! You are predicted to be free of cardiovascular disease.")
else:
    print("You are predicted to have cardiovascular disease. Please consult a doctor for further evaluation.")
```

```
             active        cardio
count  70000.000000  70000.000000
mean       0.803729      0.499700
std        0.397179      0.500003
min        0.000000      0.000000
25%        1.000000      0.000000
50%        1.000000      0.000000
75%        1.000000      1.000000
max        1.000000      1.000000
```

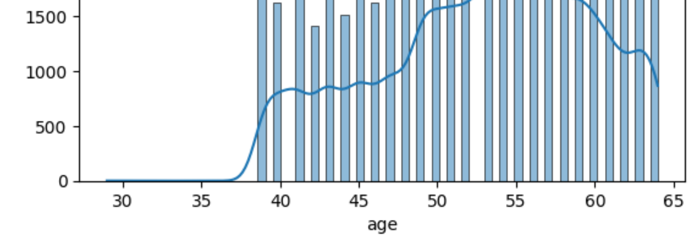


Correlation Matrix

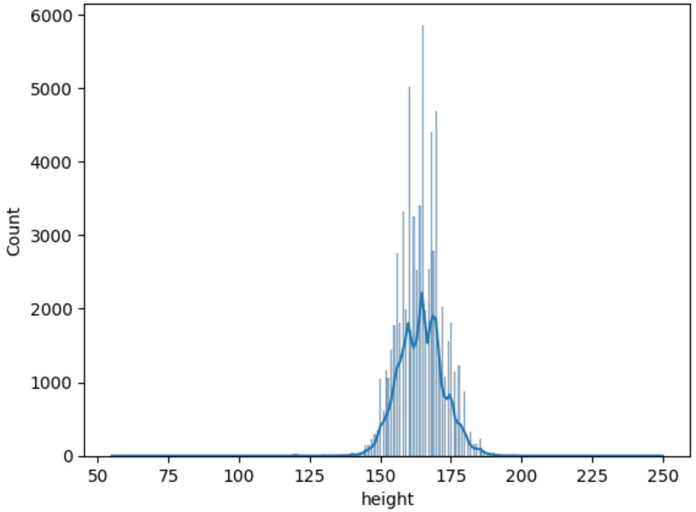
	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
age	1	-0.023	-0.082	0.054	0.021	0.018	0.15	0.098	-0.048	-0.03	-0.01	0.24
gender	-0.023	1	0.5	0.16	0.006	0.015	-0.036	-0.02	0.34	0.17	0.0059	0.0081
height	-0.082	0.5	1	0.29	0.0055	0.0062	-0.05	-0.019	0.19	0.094	-0.0066	-0.011
weight	0.054	0.16	0.29	1	0.031	0.044	0.14	0.11	0.068	0.067	-0.017	0.18
ap_hi	0.021	0.006	0.0055	0.031	1	0.016	0.024	0.012	-0.00092	0.0014	-3.3e-05	0.054
ap_lo	0.018	0.015	0.0062	0.044	0.016	1	0.024	0.011	0.0052	0.011	0.0048	0.066
cholesterol	0.15	-0.036	-0.05	0.14	0.024	0.024	1	0.45	0.01	0.036	0.0099	0.22
gluc	0.098	-0.02	-0.019	0.11	0.012	0.011	0.45	1	-0.0048	0.011	-0.0068	0.089
smoke	-0.048	0.34	0.19	0.068	-0.00092	0.0052	0.01	-0.0048	1	0.34	0.026	-0.015
alco	-0.03	0.17	0.094	0.067	0.0014	0.011	0.036	0.011	0.34	1	0.025	-0.0073
active	-0.01	0.0059	-0.0066	-0.017	-3.3e-05	0.0048	0.0099	-0.0068	0.026	0.025	1	-0.036
cardio	0.24	0.0081	-0.011	0.18	0.054	0.066	0.22	0.089	-0.015	-0.0073	-0.036	1



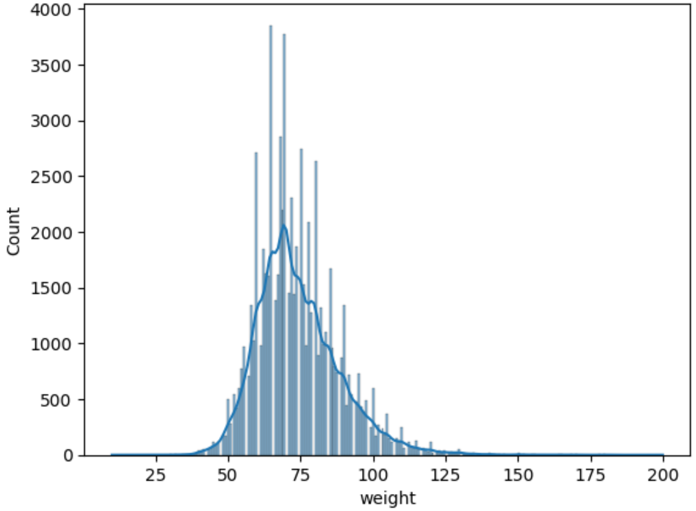
Distribution of age



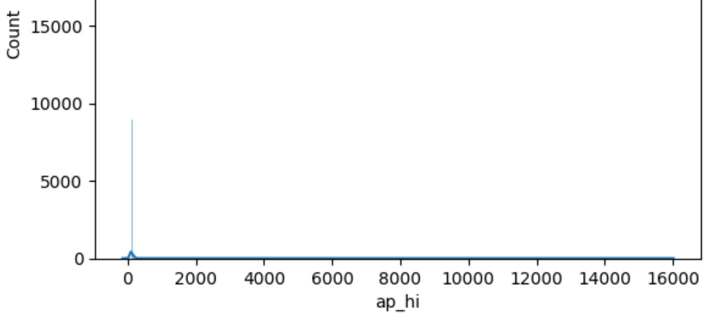
Distribution of height

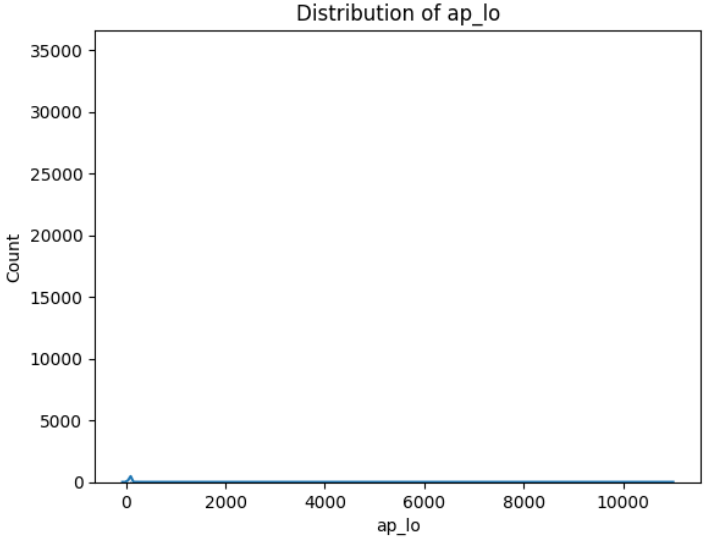


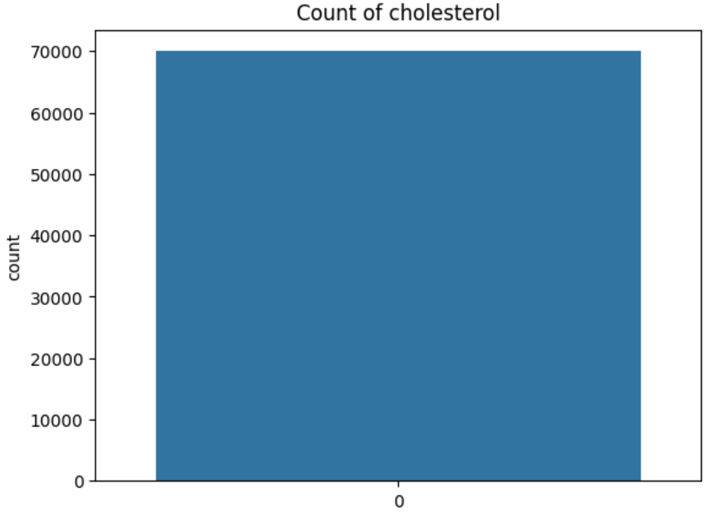
Distribution of weight

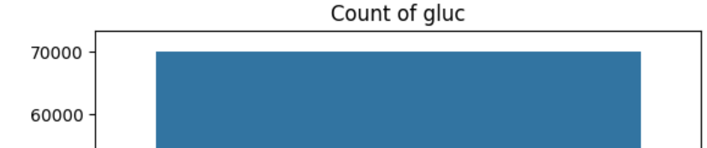


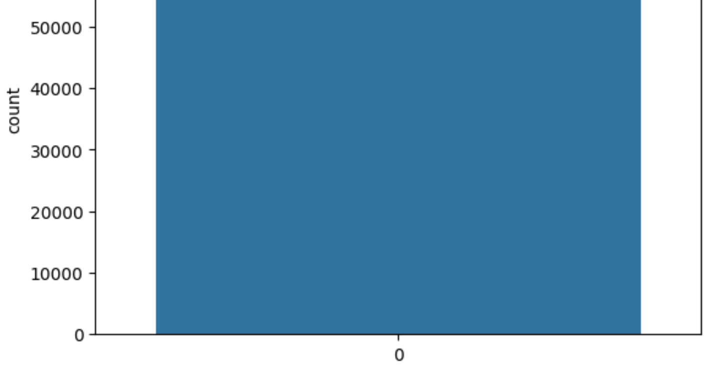
Distribution of ap_hi

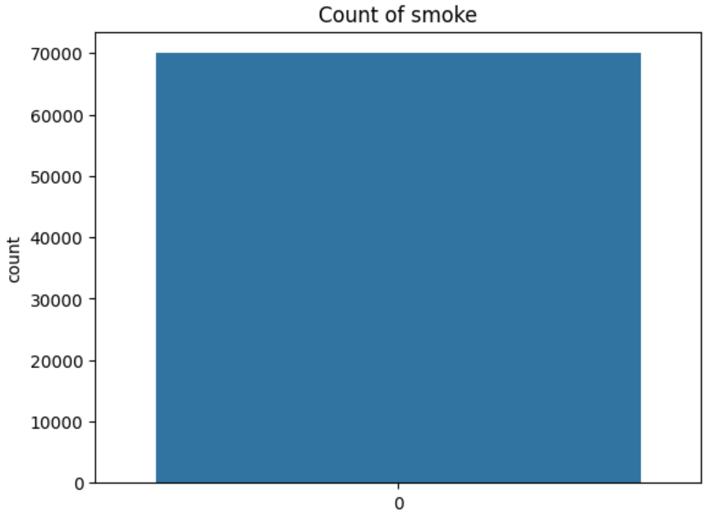

Count
15000
10000
5000
0
0
2000
4000
6000
8000
10000
12000
14000
16000
ap_hi

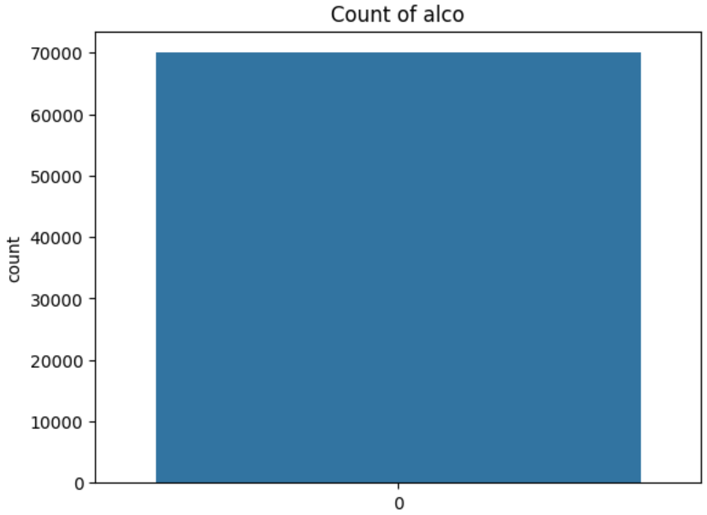

Distribution of ap_lo
35000
30000
25000
20000
15000
10000
5000
0
Count
0
2000
4000
6000
8000
10000
ap_lo

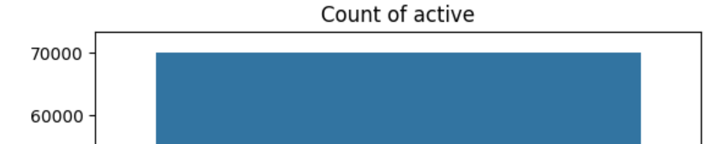

Count of cholesterol
70000
60000
50000
40000
30000
20000
10000
0
count
0

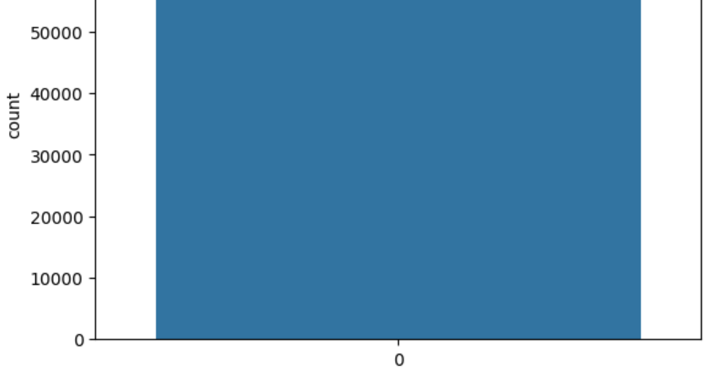

Count of gluc
70000
60000




Count of smoke


Count of alco


Count of active



```
Accuracy: 0.74

Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.78      0.75      6988
           1       0.76      0.70      0.73      7012

    accuracy                           0.74     14000
   macro avg       0.74      0.74      0.74     14000
weighted avg       0.74      0.74      0.74     14000


Confusion Matrix:
[[5444 1544]
 [2133 4879]]

Enter your age in years: 34434
Enter your gender (0: Female, 1: Male): 1
Enter your height in centimeters: 157
Enter your weight in kilograms: 45
Enter your systolic blood pressure: 56
Enter your diastolic blood pressure: 56
Enter your cholesterol level (1: normal, 2: above normal, 3: well above normal): 1
Enter your glucose level (1: normal, 2: above normal, 3: well above normal): 1
Do you smoke? (0: No, 1: Yes): 0
Do you consume alcohol? (0: No, 1: Yes): 0
Are you physically active? (0: No, 1: Yes): 0


Congratulations! You are predicted to be free of cardiovascular disease.
```