

```
In [4]: 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
import urllib.request
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
df = pd.read_csv("framingham.csv")

# Data preprocessing
df = df.dropna() # Remove rows with missing values for simplicity
X = df.drop(columns=['TenYearCHD', 'education']) # Features
y = df['TenYearCHD'] # Target variable

# Encode categorical variables (if any)
X = pd.get_dummies(X, drop_first=True)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train a Random Forest classifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train_scaled, y_train)

# Evaluate the model
y_pred = clf.predict(X_test_scaled)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# Visualizations
# Distribution of the target variable
plt.figure(figsize=(8, 6))
sns.countplot(x='TenYearCHD', data=df)
plt.title('Distribution of TenYearCHD')
plt.xlabel('TenYearCHD')
plt.ylabel('Count')
plt.show()

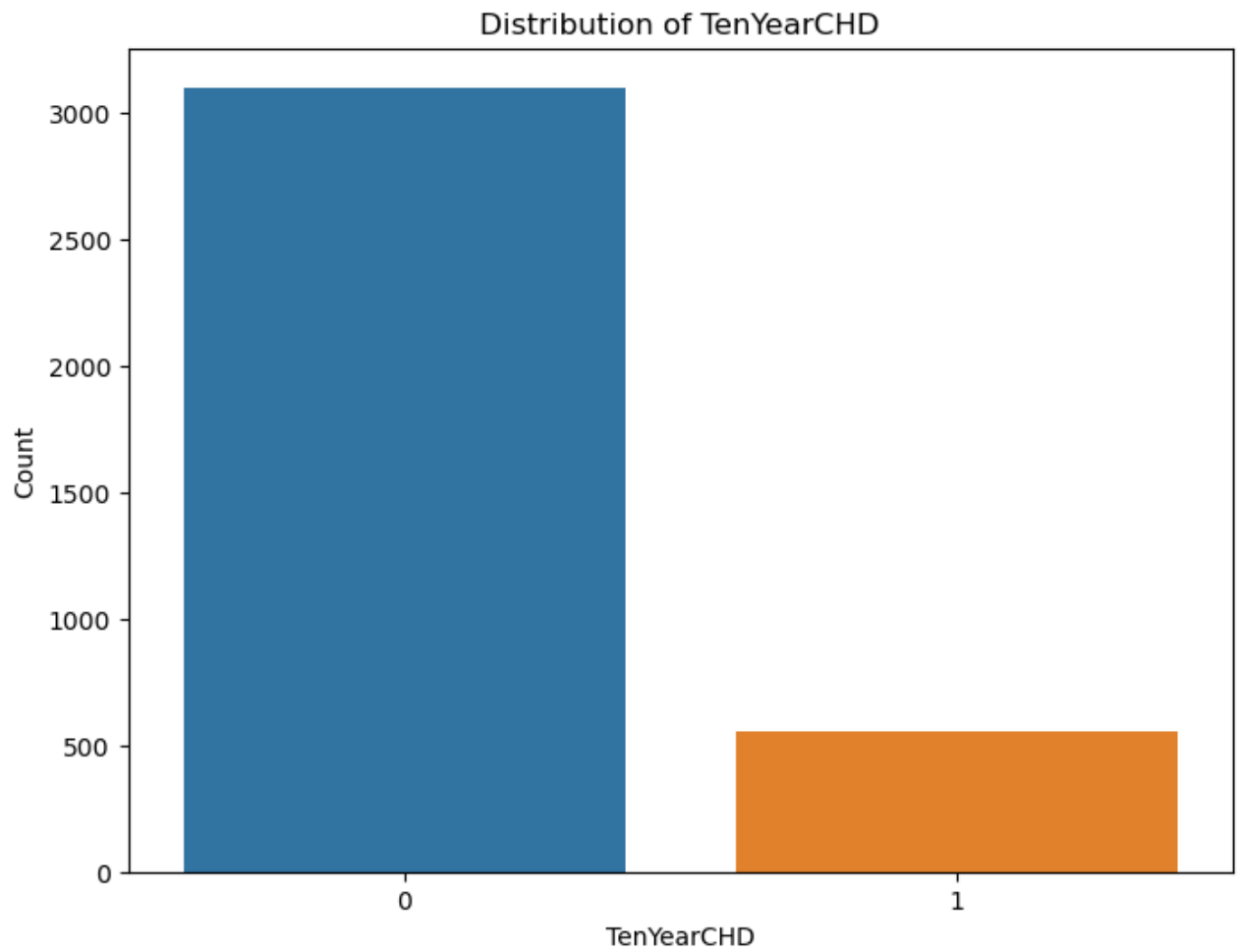
# Correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
```

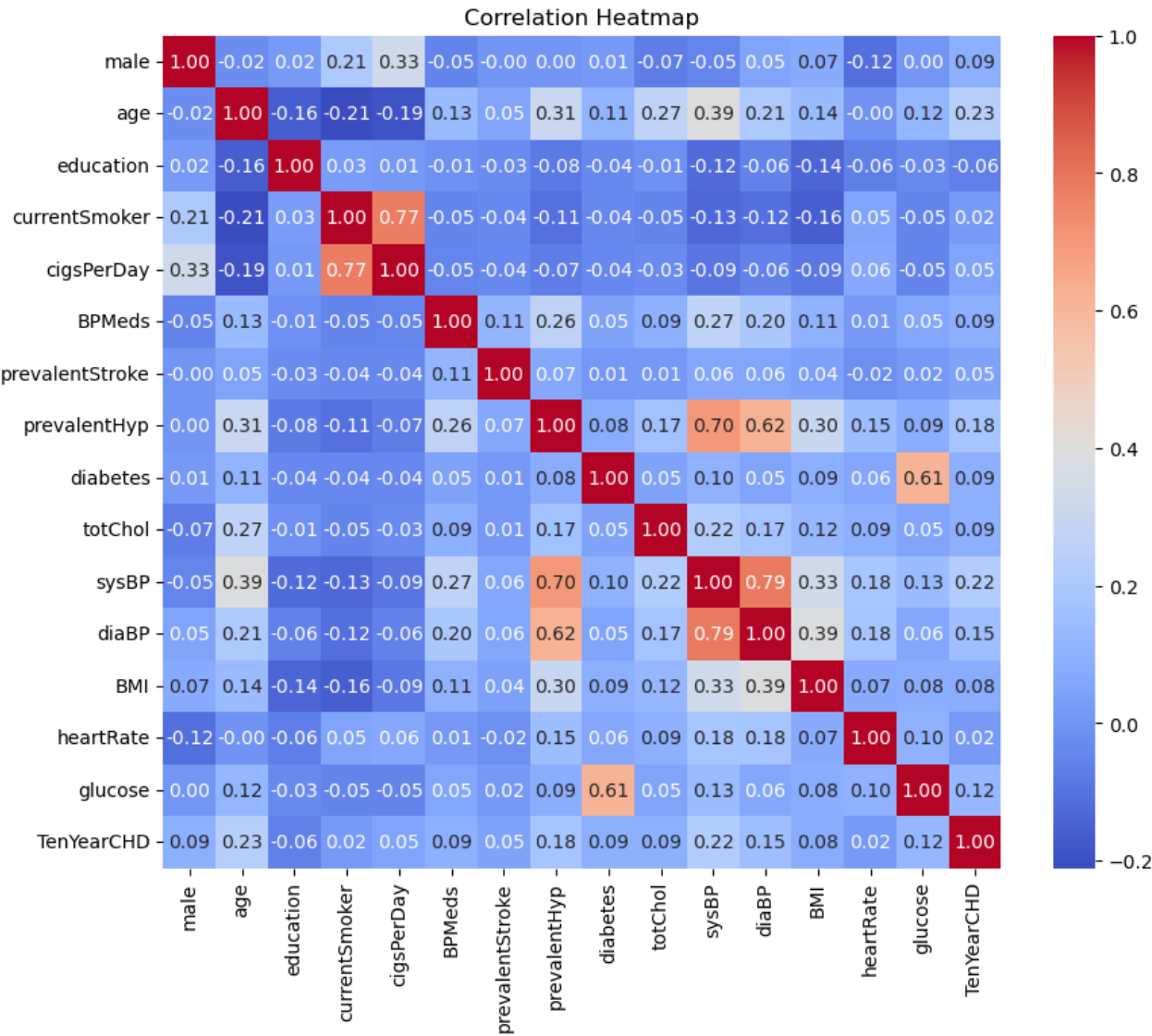
```
plt.title('Correlation Heatmap')  
plt.show()
```

Accuracy: 0.8346994535519126

Classification Report:

	precision	recall	f1-score	support
0	0.84	0.99	0.91	610
1	0.55	0.05	0.09	122
accuracy			0.83	732
macro avg	0.69	0.52	0.50	732
weighted avg	0.79	0.83	0.77	732





```
In [ ]:
```