Analyzing Cardiometabolic Risk Factors: Insights from Mutual Information

STA 160

Instructor: Fushing Hsieh

Written By: Lynn Waiyan Kyaw, Zhe Jiang, Carl Ge

1 Abstract

Cardiometabolic diseases, including heart disease, diabetes, and stroke, are leading causes of morbidity and mortality globally. This study leverages data from the Behavioral Risk Factor Surveillance System (BRFSS) to explore the relationships between various health indicators and the risk of developing these diseases. Utilizing statistical measures such as mutual information, entropy, and odds ratios, we analyzed the influence of predictors on cardiometabolic outcomes across different subpopulations. Our findings indicate that factors such as body mass index (BMI), general health (GenHlth), and high blood pressure (HighBP) consistently show strong associations with diabetes, heart disease, and stroke.

Our methodology involved defining conditional subpopulations based on the presence of one cardiometabolic disease of the two to evaluate the existence of the third disease. This approach allowed for a focused analysis on how specific predictors influence the risk of diabetes, heart disease, and stroke within these subpopulations. After the implementaion, we found that general health(GenHlth) is the top factor across all conditions and diseases, indicating a potential public health interventions to mitigate the impact of cardiometabolic diseases.

2 Introduction

Cardiometabolic diseases such as heart disease remain a formidable health challenge globally, with the United States experiencing a significant impact each year. According to the Centers for Disease Control and Prevention (CDC), the U.S. consistently ranks among the countries most affected by heart diseases, reflecting a health concern that affects millions of individuals in the country. Annually, heart disease claims a staggering number of lives in the U.S., with approximately 695,000 deaths attributed to it in 2021, making it the leading cause of mortality nationwide for men, women, and people of most racial and ethnic groups [1]. In terms of other cardiometabolic diseases, recent projections from a comprehensive study published in the Journal of the American College of Cardiology shed light on the escalating trajectory of cardiovascular diseases (CVD) in the United States. The study, conducted using data from the 2020 U.S. Census Bureau and the National Health and Nutrition Examination Survey, offers an outlook on the future risks of CVDs across different demographic groups. By analyzing trends from 2025 to 2060, the study highlights a trend of rising cardiovascular risk factors and diseases. Among the general population, rates of key risk factors such as diabetes, hypertension, dyslipidemia, obesity, strokes are projected to surge significantly [2]. The primary goal of analyzing both the overall population and specific subpopulations(introduced in later section) is to understand how disease dynamics or risk factor distributions differ among varied demographic or clinical groups. This approach enables us to analyze both the overall population and specific subpopulations, providing insights into how disease dynamics or risk factor distributions vary among different demographic or clinical groups. Additionally, we will compare rankings of mutual information to those of the overall population if there is a specific pattern for each disease.

3 Dataset Description

The heart disease indicators dataset that was utilized in this study was obtained from the Kaggle library: https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset/data [3]. To summarize, this dataset is derived from the Behavioral Risk Factor Surveillance System (BRFSS), an annual health-related telephone survey conducted by the CDC since 1984. The dataset for the year 2015 comprises responses from 441,455 individuals with 330 features, including questions directly asked of participants and calculated variables. After cleaning, the dataset contains 253,680 survey responses, with a notable class imbalance: 229,787 respondents do not have or have not had heart disease, while 23,893 have reported heart disease. In terms of the variables included in the dataset, there is one binary target variable: HeartDiseaseorAttack and 21 feature variables (e.g. Education, Income, BMI, PhysActivity) that are either binary or ordinal.

3.1 Dataset Cleaning and Preparation

Before working on the heart disease indicators dataset, we performed a data cleaning process to increase the accuracy and reliability of the dataset. First, we searched the dataset for any missing values since they can skew the results of the correlation analysis and may lead to inaccurate interpretations. After checking the dataset for missing values, we found that there were no missing values. Then, there are 253680 total observations for this dataset.

We then also divided the variable types into binary and non-binary categories. Binary predictor variables are: highBP, HighChol, CholCheck, DiffWalk, Sex, Smoker, PhysActivity, Fruits, Veggies, HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, Stroke, HeartDiseaseorAttack, Diabetes(we combine 1 and 2 to 1, so finally it has 2 values: 0 and 1). The non-binary ones are: GenHlth, MentHlth, PhysHlth, Age, Education, Income, BMI. The stats summary of each variable are shown below:

Table 1: Summary Statistics of Health Indicators

Variable	**Mean**	**Standard Deviation**
Heart Disease or Attack	0.094186	0.292087
High BP	0.429001	0.494934
High Chol	0.424121	0.494210
Chol Check	0.962670	0.189571
BMI	28.382364	6.608694
Smoker	0.443169	0.496761
Stroke	0.040571	0.197294
Diabetes	0.296921	0.698160
Physical Activity	0.756544	0.429169
Fruits	0.634256	0.481639
Veggies	0.811420	0.391175
Heavy Alcohol Consumption	0.056197	0.230302
Any Healthcare	0.951053	0.215759
No Doc because Cost	0.084177	0.277654
General Health	2.511392	1.068477
Mental Health	3.184772	7.412847
Physical Health	4.242081	8.717951
Difficulty Walking	0.168224	0.374066
Sex	0.440342	0.496429
Age	8.032119	3.054220
Education	5.050434	0.985774
Income	6.053875	2.071148

Next, we conducted a correlation analysis by producing a correlation matrix for our variables. A correlation matrix is a useful tool in data analysis as it summarizes the relationships between variables within a dataset. This information is invaluable for identifying patterns, assessing the degree of independence or dependence among variables, selecting relevant features for analysis, and prioritizing variables based on their correlations with the target outcome (diabetes, stroke, and heart disease).

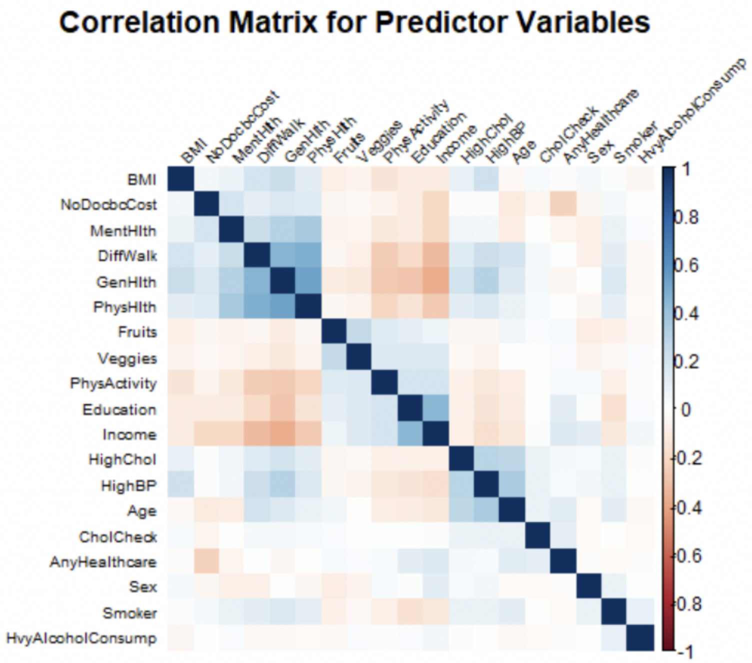


Figure 1: Predictors Correlation Matrix

In this predictors correlation matrix, we can see that blue shading indicates a positive correlation, suggesting a moderate to high linear relationship where increases in one variable are associated with corresponding increases in another. Light orange areas represent moderate negative correlations, indicating a more moderate but still noticeable relationship between variables. On the other hand, white shading signifies no correlation or very weak correlations, suggesting little to no linear relationship between the variables. Using the information from the correlation matrix however, we can see that only a few predictors exhibit strong positive correlations (darker blue areas) as they indicate potential dependencies, and most predictors have areas with no or weak correlations (white areas) suggest potential mutual independence between these variables.

After that we divided the responses (Stroke, HeartDiseaseorAttack, Diabetes) into 8 categories using 3 numbers, following the order above) to represent the response feature. For example, 1-0-1 represents the subpopulation has Stroke and Diabetes, but do not have HeartDisease or Attack.

4 Methodology

Entropy: we used entropy to measure uncertainty or randomness in our data. By calculating entropy for the response variables (diabetes, stroke, and heart disease), we can quantify the amount of information or predictability associated with each variable. The formula that we used for calculating entropy is:

$$H(X) = -\sum_{i=1}^{n} P(X_i) \log_2 P(X_i)$$

Conditional Entropy: we leveraged conditional entropy (CE) to evaluate the relationship between the response variables (diabetes, stroke, and heart disease) and predictor in the dataset. The formula for the conditional entropy is given by:

$$H(Y \mid X) = -\sum_{x \in X} \sum_{y \in Y} P(x, y) \log_2 \frac{P(x, y)}{P(x)}$$

Where Y represents the response variables and X is the fused variable represented as $X = (X_1, X_2)$.

We calculated $CE[Y \mid X]$ to quantify the uncertainty in each response variable given the knowledge of the predictor X. Additionally, if X is a fused variable, such as $X = (X_1,\ X_2)$, we compared $CE[Y] - CE[Y \mid X]$ with $CE[Y] - CE[Y \mid X_1] + CE[Y] - CE[Y \mid X_2]$ to determine the presence of interacting effects. Lower values of conditional entropy indicate stronger associations or dependencies between the response variables and the fused variable, highlighting their mutual influence and the potential for joint effects in our analysis.

Mutual Information: We calculated the mutual information between X and Y by:

$$I[X, Y] = CE[Y] - CE[Y \mid X] = CE[X] - CE[X \mid Y]$$

We calculated the mutual information between fused variable (X_1, X_2) and the response variable Y using the formula:

$$I[(X_1, X_2), Y] = CE[Y] - CE[Y \mid X_1] + CE[Y] - CE[Y \mid X_2] + I[(X_1, X_2) \mid Y] - I[X_1, X_2]$$

Where $I[(X_1, X_2), Y]$ represents the mutual information between non-fused covariates X_1 and X_2 with respect to Y, $CE[Y]$ is the conditional entropy of Y, and $CE[Y \mid (X_1, X_2)]$ is the conditional entropy of Y given the knowledge of both X_1 and X_2.

The ecological effect (I[(X1, X2), Y]) indicates the combined impact of X1 and X2 on Y, beyond their individual effects. This formula allows us to evaluate the relationships between features and the response variable, providing insights into feature importance and selection for our analysis of mutual independence or dependence between diabetes, stroke, and heart disease. Utilizing this methodology aids in identifying influential predictors and enhances our understanding of predictive factors in our cardiometabolic dataset.

Odds Ratio: We utilized odds ratios as a statistical measure to assess the relationship between our response variables and our predictor variables, aiding in the assessment of their mutual independence or dependence. The formula of the odds ratio is given by:

$$\text{Odds Ratio} = \frac{A/C}{B/D}$$

Table 2: Odds Ratio Table

	Outcome Yes	**Outcome No**
Predictor Yes	A	B
Predictor No	C	D

It is important to note that odds ratios are applicable only to binary variables, and we computed them based on contingency tables, which helped quantify the likelihood of an event (such as disease occurrence) given the presence or absence of another condition (e.g., health behavior).

5 Results & Discussion

Firstly, we calculate the odds ratio for all the binary predictors with respect to each individual response.

Table 3: Odds Ratio Between Health Indicators and Diseases

Predictor	Stroke	Heart Disease or Attack	Diabetes
HighBP	4.02	4.59	4.78
HighChol	2.58	3.59	3.24
CholCheck	2.60	3.64	5.87
Smoker	1.86	2.20	1.41
PhysActivity	0.49	0.54	0.50
Fruits	0.87	0.87	0.79
Veggies	0.62	0.73	0.68
HvyAlcoholConsump	0.64	0.59	0.41
AnyHealthcare	1.25	1.40	1.21
NoDocbcCost	1.69	1.41	1.41
DiffWalk	5.24	4.27	3.70
Sex	1.03	1.80	1.18

When examining the overall odds ratios, we observed notable differences compared to the results obtained from mutual information analysis(figure 2, 4, 7) for the overall population. Due to these inconsistencies, we have opted to rely solely on mutual information for analysis within the subpopulation. The decision to prioritize mutual information over odds ratios stems from specific observations, particularly highlighted in the analysis of the effect of CholCheck on Diabetes within the subpopulation. Despite the high odds ratio associated with cholesterol tests among diabetic patients, signifying a frequent occurrence of cholesterol tests among this group, the corresponding mutual information value is low. This discrepancy suggests that undergoing a cholesterol test may not significantly influence the development of diabetes in patients. Additionally, high conditional entropy (1.92815) and low mutual information (0.00686) between CholCheck and diabetes suggest that while cholesterol tests (CholCheck) are widely performed among diabetic patients, they have little direct impact on the development of diabetes. Therefore, we decided to mainly use mutual information only for the rest of the analysis since mutual information covers a more complex association compared to the odds ratio(only linear).

Conditioned Subpopulation:

We structured our analysis by defining subpopulations based on the exclusive presence of one of two diseases, either Stroke or Heart Disease, specifically excluding cases where both diseases occur simultaneously. This method helps us focus on how the presence of one condition influences the development of another condition, considered as target. For example, when studying the influence of predictors on Diabetes, we consider subpopulations that include individuals diagnosed with either Stroke or Heart Disease alone. Specifically, these subpopulations are categorized as follows: having Stroke but not Heart Disease or having Heart Disease but not Stroke. This division allows us to isolate the effects of other diseases(response) on targeted disease(response), allowing us to explore the association between predictors and response in a more concise way.

Diabetes:

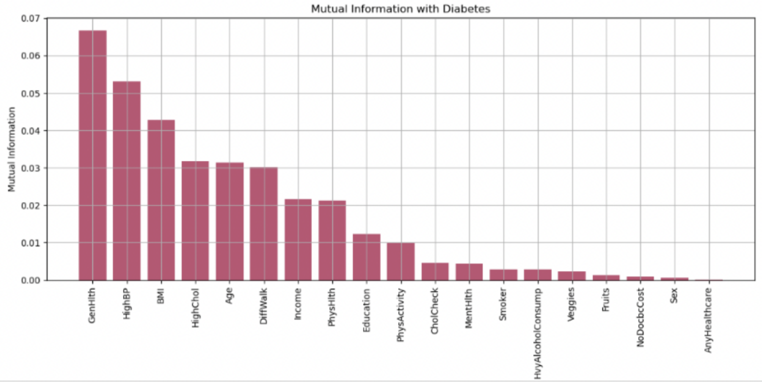


Figure 2: Mutual Information Histogram for Diabetes (Overall)

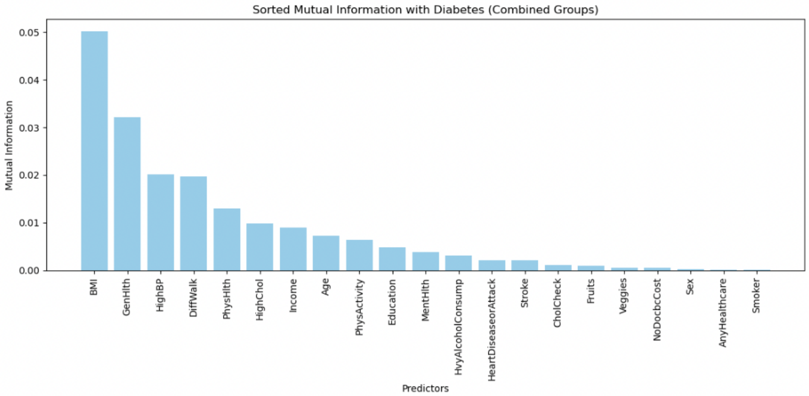


Figure 3: Mutual Information Histogram for Diabetes (Subpopulation)

Starting with our first subdivision group,target on diabetes, we noticed that the top features of diabetes in both the overall population and the subpopulations are the patient's body mass index (BMI), general health (GenHlth), and high blood pressure (HighBP) according to Figure 2 and Figure 3. The consistency across groups suggests that these factors are robust features of diabetes risk, irrespective of other cardiovascular conditions. Moreover, the mutual information for BMI is the subpopulation has been significantly increased compared to the overall dataset, indicating that BMI plays a more important role in diabetes risk than individuals who are already suffering from either stroke or heart disease.

Heart Disease or Attack:

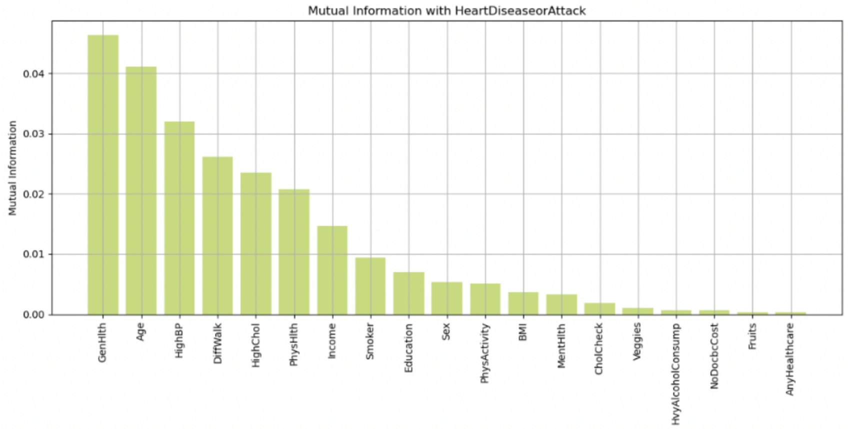


Figure 4: Mutual Information Histogram for Heart Disease or Attack (Overall)

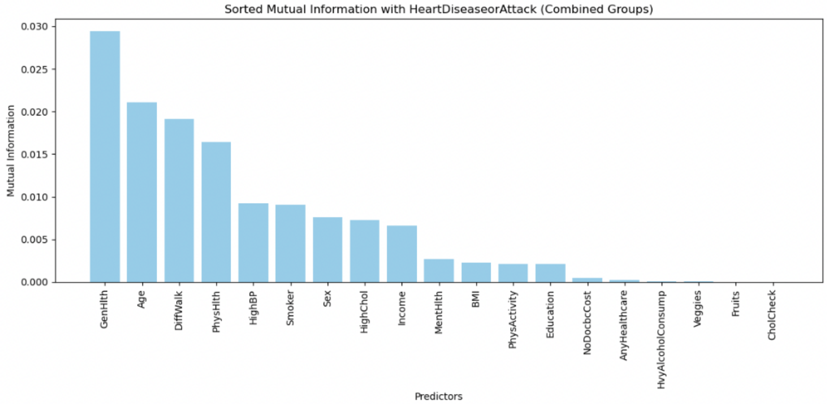


Figure 5: Mutual Information Histogram for Heart Disease or Attack (Subpopulation)

Onto the next subdivision group targeting on heart disease, we can dicover that general health (GenHlth), difficulty-with-walking (DiffWalk), and patient age (Age) emerge as the top recurring highest features in both the overall and subpopulation plot. This indicates that these factors are influential across all conditions.

However, we noticed something counter-intuitive where the mutual information score for HighBP were high in the overall population in Figure 4, but it is relatively low in the subpopulation plot in Figure 5 because in terms of medical context, high blood should both have positive associations with heart disease or attack. To study this unusual strong occurrence, we decided to investigate the interaction effect of other factor with highBP to heart disease or attack.

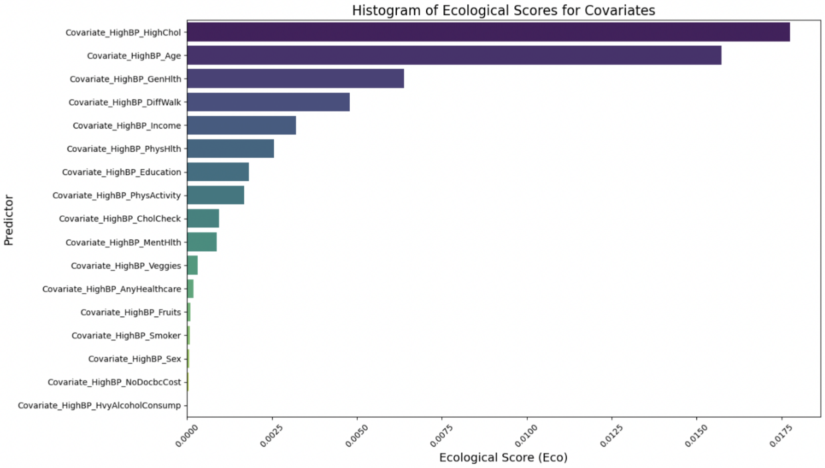


Figure 6: Ecological Scores Histogram

The investigation of interaction between HighBP and HighChol revealed interesting findings, concerning the unusual occurrence from the plots displayed earlier. Initially, the individual Mutual Information (MI) values for high cholesterol and high blood pressure were observed to fall within the middle range of the overall MI values concerning heart disease response. However, when hypertension (HighBP) and high cholesterol (HighChol) were considered as fused variables, the resulting ecological scores surpassed the individual MI values, indicating a strong interaction effect. This interaction effect suggests that the presence of one condition (high cholesterol) amplifies the impact of the other (high blood pressure) on the risk of heart disease. Consequently, when high blood pressure is coupled with high cholesterol, the risk of developing heart disease escalates significantly. This particular investigation explores how high blood pressure (HighBP) and high cholesterol (HighChol) interact to amplify the risk of heart disease, when considered together.

Stroke

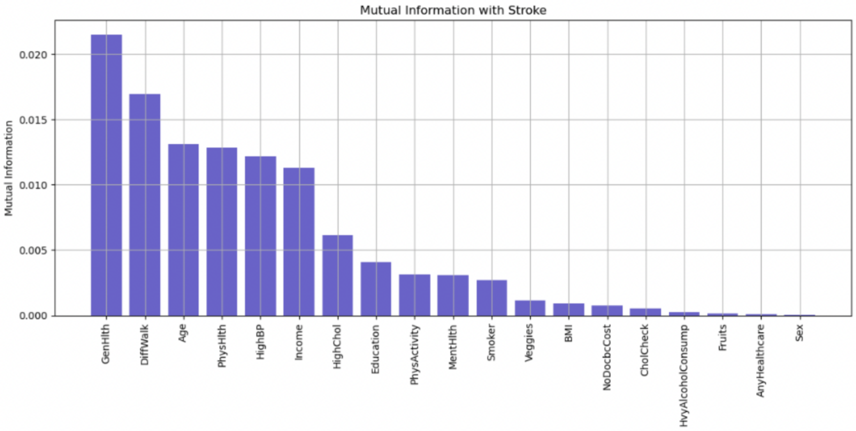


Figure 7: Mutual Information Histogram for Stroke (Overall)

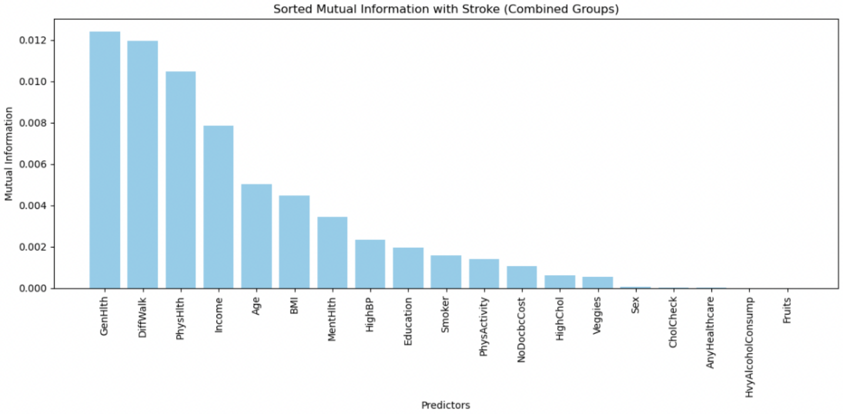


Figure 8: Mutual Information Histogram for Stroke (Subpopulation)

For our last subdivision targeting on stroke, we can see on Figure 7 and Figure 8 that general health (GenHlth), difficulty-with-walking (DiffWalk), and physical health (PhysHlth) are the common highest features in both the overall and subpopulation plot. Although the ranks of the features varies, but in general, the ranks are not too far away to their original positions.

6 Conclusion

To summarize our findings, the consistent patterns observed in the mutual information rankings across subpopulations indicate a robust association between certain predictors and cardiometabolic diseases, regardless of the presence of other conditions.

Similar patterns of mutual information across these subpopulations indicate that certain predictors are robustly associated with a cardiodisease irrespective of the presence of one of the other two diseases. In this case, general health's consistent ranking as a top predictor in both the overall population and subpopulations suggests it holds a central role in predicting the risk of these major health conditions. This implies a strong association between overall perceived health status and the risk of developing stroke, diabetes, and heart disease, suggesting that interventions targeting general health are likely to be effective in managing diabetes risk universally. Based on that, we can come up with some strategies based on GenHlth for preventing cardiodisease. These strategies could include promoting healthy lifestyle changes, such as increased physical activity and balanced diets, which are universally beneficial for good general health, resulting in a good intervention of cardiodisease.

7 References

1. Centers for Disease Control and Prevention. (2024). Heart disease facts. Centers for Disease Control and Prevention. https://www.cdc.gov/heartdisease/facts.html

2. Roth, S. (2022, August 1). New US population study projects steep rise in cardiovascular diseases by 2060. American College of Cardiology. https://www.acc.org/About-ACC/Press-Releases/2022/08/01/16/37/New-US-Population-Study-Projects-Steep-Rise-in-Cardiovascular-Diseases-by-2060

3. Teboul, A. Heart disease health indicators dataset. Kaggle. https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset/data

A Appendix

Code appendix

```r
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
library(corrplot)
library(ggplot2)
library(reshape2)
library(DescTools)
library(infotheo)
library(epitools)
data = read.csv("heart_disease_health_indicators_BRFSS2015.csv", fileEncoding = "UTF-8")
head(data)

# Search for missing Values

missing_values <- colSums(is.na(data))

print(missing_values) # No missing values
responses <- data[,c("HeartDiseaseorAttack", "Stroke", "Diabetes")]
predictors <- data[,!(names(data) %in% c("HeartDiseaseorAttack", "Stroke", "Diabetes"))]
corr_responses <- cor(responses, use = "complete.obs")
corr_predictors <- cor(predictors, use = "complete.obs")

corrplot(corr_responses, method = "color", type = "upper", order = "hclust",
         addCoef.col = "black", tl.col = "black", tl.srt = 45, tl.cex = 0.6,
         diag = FALSE, cl.ratio = 0.1, cl.cex = 0.75,
         title = "Correlation Matrix for Response Variables")

corrplot(corr_predictors, method = "color", order = "hclust",
         tl.col = "black", tl.srt = 45, tl.cex = 0.6,
         cl.ratio = 0.1, cl.cex = 0.75,
         title = "Correlation Matrix for Predictor Variables")

corr_all <- cor(data, use = "complete.obs")

corrplot(corr_all, method = "color", order = "hclust",
         tl.col = "black", tl.srt = 45, tl.cex = 0.6,
         cl.ratio = 0.1, cl.cex = 0.75,
         title = "Correlation Matrix for Entire Dataset")
```

python_version_project

May 12, 2024

1 160 project

[ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:

```
data = pd.read_csv("heart_disease_health_indicators_BRFSS2015.csv")
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253680 entries, 0 to 253679
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   HeartDiseaseorAttack  253680 non-null  float64
 1   HighBP                253680 non-null  float64
 2   HighChol              253680 non-null  float64
 3   CholCheck             253680 non-null  float64
 4   BMI                   253680 non-null  float64
 5   Smoker                253680 non-null  float64
 6   Stroke                253680 non-null  float64
 7   Diabetes              253680 non-null  float64
 8   PhysActivity          253680 non-null  float64
 9   Fruits                253680 non-null  float64
 10  Veggies               253680 non-null  float64
 11  HvyAlcoholConsump     253680 non-null  float64
 12  AnyHealthcare         253680 non-null  float64
 13  NoDocbcCost           253680 non-null  float64
 14  GenHlth               253680 non-null  float64
 15  MentHlth              253680 non-null  float64
 16  PhysHlth              253680 non-null  float64
 17  DiffWalk              253680 non-null  float64
 18  Sex                   253680 non-null  float64
 19  Age                   253680 non-null  float64
 20  Education             253680 non-null  float64
 21  Income                253680 non-null  float64
dtypes: float64(22)
memory usage: 42.6 MB
```

```
[ ]: data['Diabetes'] = data['Diabetes'].replace(2, 1)
```

```
[ ]: data.describe()
```

```
[ ]:        HeartDiseaseorAttack         HighBP       HighChol      CholCheck  \
     count         253680.000000  253680.000000  253680.000000  253680.000000
     mean               0.094186       0.429001       0.424121       0.962670
     std                0.292087       0.494934       0.494210       0.189571
     min                0.000000       0.000000       0.000000       0.000000
     25%                0.000000       0.000000       0.000000       1.000000
     50%                0.000000       0.000000       0.000000       1.000000
     75%                0.000000       1.000000       1.000000       1.000000
     max                1.000000       1.000000       1.000000       1.000000

                      BMI         Smoker         Stroke       Diabetes  \
     count  253680.000000  253680.000000  253680.000000  253680.000000
     mean       28.382364       0.443169       0.040571       0.157588
     std         6.608694       0.496761       0.197294       0.364355
     min        12.000000       0.000000       0.000000       0.000000
     25%        24.000000       0.000000       0.000000       0.000000
     50%        27.000000       0.000000       0.000000       0.000000
     75%        31.000000       1.000000       0.000000       0.000000
     max        98.000000       1.000000       1.000000       1.000000

             PhysActivity         Fruits  …  AnyHealthcare    NoDocbcCost  \
     count  253680.000000  253680.000000  …  253680.000000  253680.000000
     mean        0.756544       0.634256  …       0.951053       0.084177
     std         0.429169       0.481639  …       0.215759       0.277654
     min         0.000000       0.000000  …       0.000000       0.000000
     25%         1.000000       0.000000  …       1.000000       0.000000
     50%         1.000000       1.000000  …       1.000000       0.000000
     75%         1.000000       1.000000  …       1.000000       0.000000
     max         1.000000       1.000000  …       1.000000       1.000000

                  GenHlth       MentHlth       PhysHlth       DiffWalk  \
     count  253680.000000  253680.000000  253680.000000  253680.000000
     mean        2.511392       3.184772       4.242081       0.168224
     std         1.068477       7.412847       8.717951       0.374066
     min         1.000000       0.000000       0.000000       0.000000
     25%         2.000000       0.000000       0.000000       0.000000
     50%         2.000000       0.000000       0.000000       0.000000
     75%         3.000000       2.000000       3.000000       0.000000
     max         5.000000      30.000000      30.000000       1.000000

                      Sex            Age      Education         Income
     count  253680.000000  253680.000000  253680.000000  253680.000000
     mean        0.440342       8.032119       5.050434       6.053875
```

```
std         0.496429      3.054220      0.985774      2.071148
min         0.000000      1.000000      1.000000      1.000000
25%         0.000000      6.000000      4.000000      5.000000
50%         0.000000      8.000000      5.000000      7.000000
75%         1.000000     10.000000      6.000000      8.000000
max         1.000000     13.000000      6.000000      8.000000

[8 rows x 22 columns]
```

Overall Odds Ratio

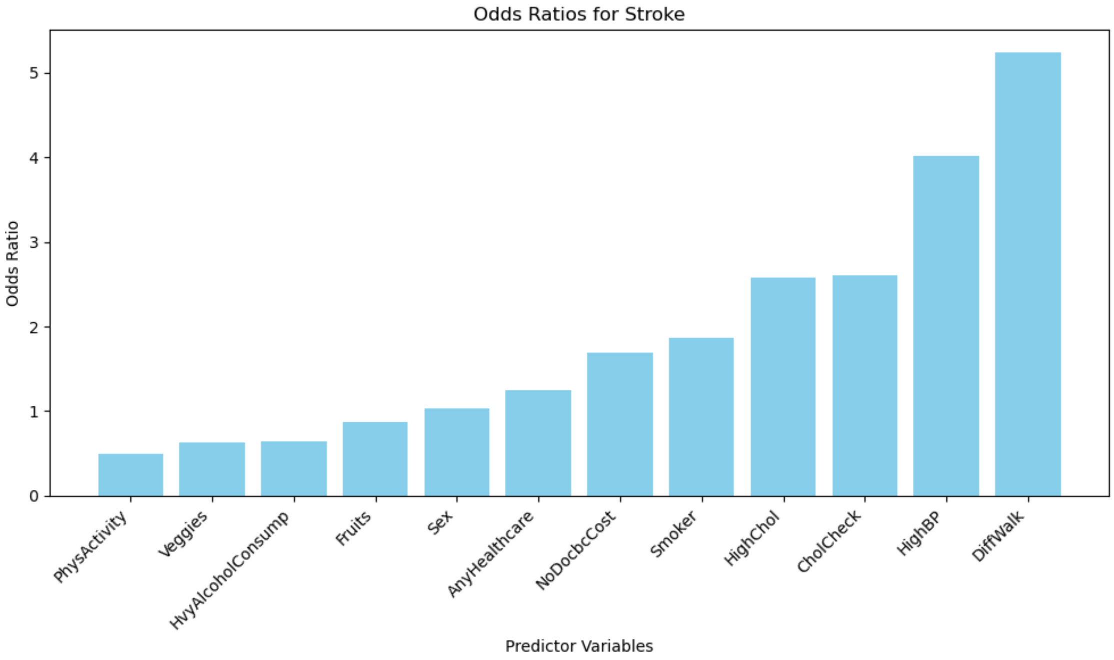
[ ]:

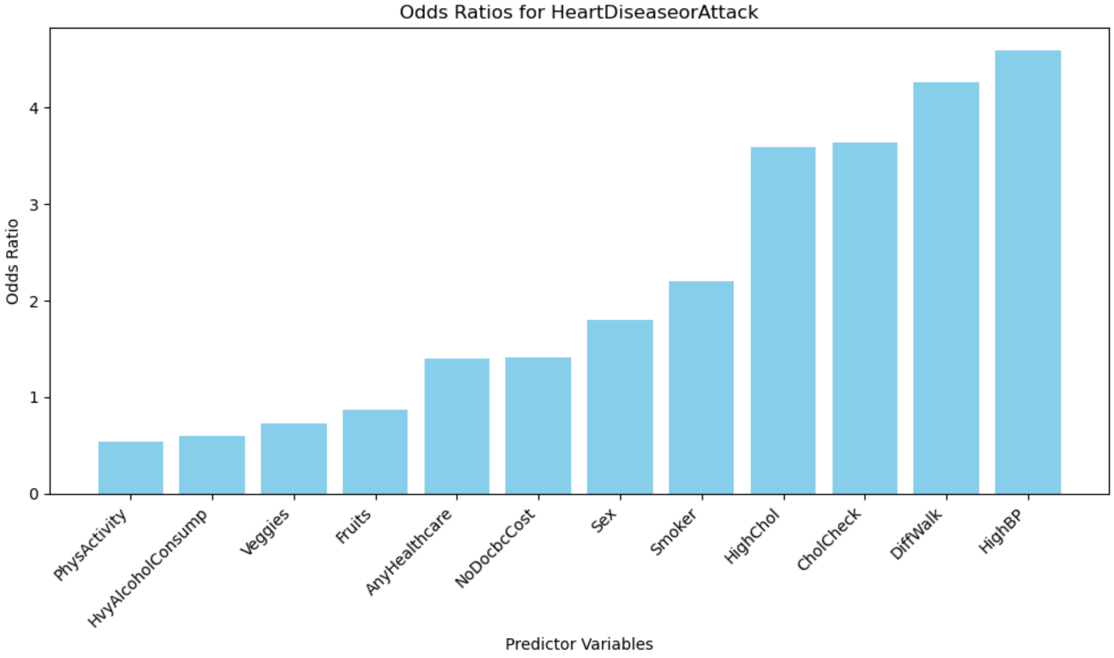
```python
def calculate_odds_ratio(Y, X):
    contingency_table = pd.crosstab(Y, X)
    if contingency_table.shape != (2, 2):
        return None
    a = contingency_table.iloc[0, 0]
    b = contingency_table.iloc[0, 1]
    c = contingency_table.iloc[1, 0]
    d = contingency_table.iloc[1, 1]
    if b == 0 or c == 0:
        return float('inf')
    return (a * d) / (b * c)

def calculate_odds_ratios(data, response_vars, predictor_vars):
    odds_ratios = pd.DataFrame(index=response_vars, columns=predictor_vars)
    for response in response_vars:
        for predictor in predictor_vars:
            if predictor != response:
                try:
                    odds_ratio = calculate_odds_ratio(data[response],␣
 ↪data[predictor])
                    odds_ratios.loc[response, predictor] = odds_ratio
                except Exception as e:
                    odds_ratios.loc[response, predictor] = None
    return odds_ratios.apply(pd.to_numeric, errors='coerce')

def plot_sorted_odds_ratios(odds_ratios, response_vars):
    for response in response_vars:
        sorted_odds = odds_ratios.loc[response].dropna().replace(float('inf'),␣
 ↪None).dropna().sort_values()
        if not sorted_odds.empty:
            plt.figure(figsize=(10, 6))
            plt.bar(sorted_odds.index, sorted_odds.values, color='skyblue')
            plt.xlabel('Predictor Variables')
            plt.ylabel('Odds Ratio')
            plt.title(f'Odds Ratios for {response}')
            plt.xticks(rotation=45, ha='right')
            plt.tight_layout()
            plt.show()

def process_data_and_plot(data, response_vars, predictor_vars):
    odds_ratios = calculate_odds_ratios(data, response_vars, predictor_vars)
    plot_sorted_odds_ratios(odds_ratios, response_vars)


response_vars = ["Stroke", "HeartDiseaseorAttack", "Diabetes"]
predictor_vars = ['HighBP', 'HighChol', 'CholCheck', 'Smoker', 'PhysActivity',␣
 ↪'Fruits',
                  'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare',␣
 ↪'NoDocbcCost',
                  'DiffWalk', 'Sex']
```

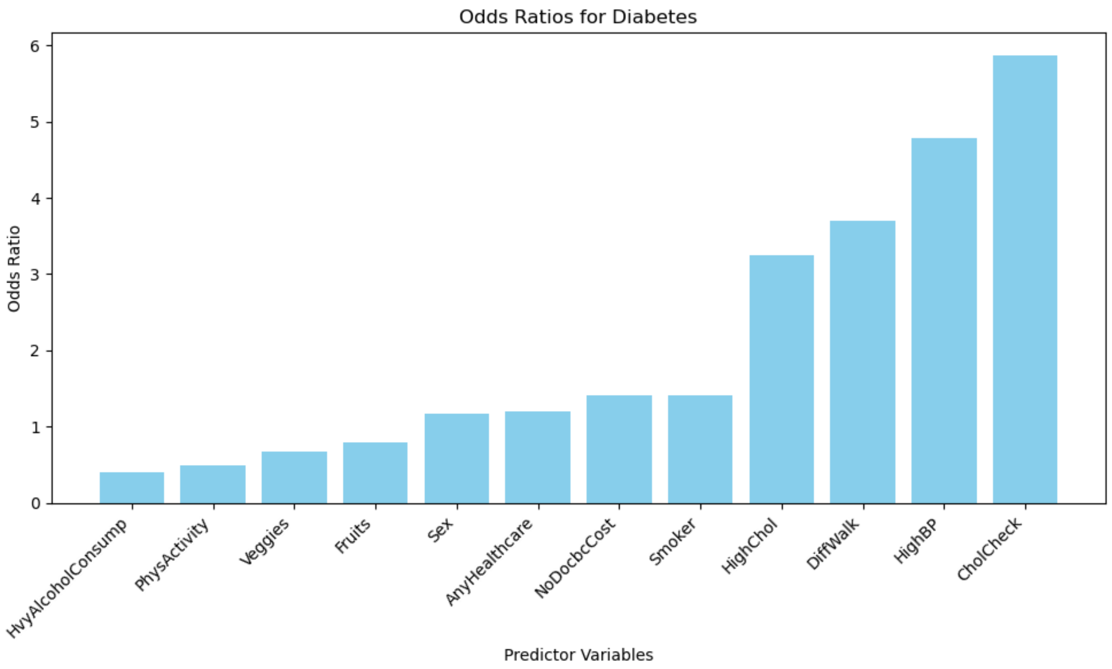
[ ]:
```
calculate_odds_ratios(data, response_vars, predictor_vars)
```

[ ]:
```
                        HighBP  HighChol  CholCheck    Smoker  PhysActivity
Fruits   Veggies  HvyAlcoholConsump  AnyHealthcare  NoDocbcCost  DiffWalk
Sex
Stroke                4.016704  2.583564   2.602611  1.861800      0.491360
0.870494  0.624904           0.639454       1.254037     1.685910  5.239774
1.030826
HeartDiseaseorAttack  4.592099  3.589073   3.635014  2.203943      0.535980
0.870471  0.727845           0.593841       1.400159     1.407146  4.266085
1.803161
Diabetes              4.781584  3.241590   5.868415  1.410944      0.495664
0.790307  0.680355           0.405188       1.209389     1.408323  3.695512
1.176812
```

[ ]:
```
process_data_and_plot(data, response_vars, predictor_vars)
```


Odds Ratios for Stroke
Odds Ratio
0
1
2
3
4
5
PhysActivity
Veggies
HvyAlcoholConsump
Fruits
Sex
AnyHealthcare
NoDocbcCost
Smoker
HighChol
CholCheck
HighBP
DiffWalk
Predictor Variables


Odds Ratios for HeartDiseaseorAttack
Odds Ratio
0
1
2
3
4
PhysActivity
HvyAlcoholConsump
Veggies
Fruits
AnyHealthcare
NoDocbcCost
Sex
Smoker
HighChol
CholCheck
DiffWalk
HighBP
Predictor Variables



Entropy and MI

[ ]:

```python
def entropy(X):
    unique, count = np.unique(X, return_counts=True, axis=0)
    prob = count/len(X)
    en = np.sum((-1)*prob*np.log2(prob))
    return en

def jEntropy(X,Y):
    XY = np.c_[X,Y]
    return entropy(XY)

def cEntropy(X,Y):
    return jEntropy(X,Y) - entropy(Y)

def calculate_mi(X,Y):
    return entropy(X) - cEntropy(X,Y)
```

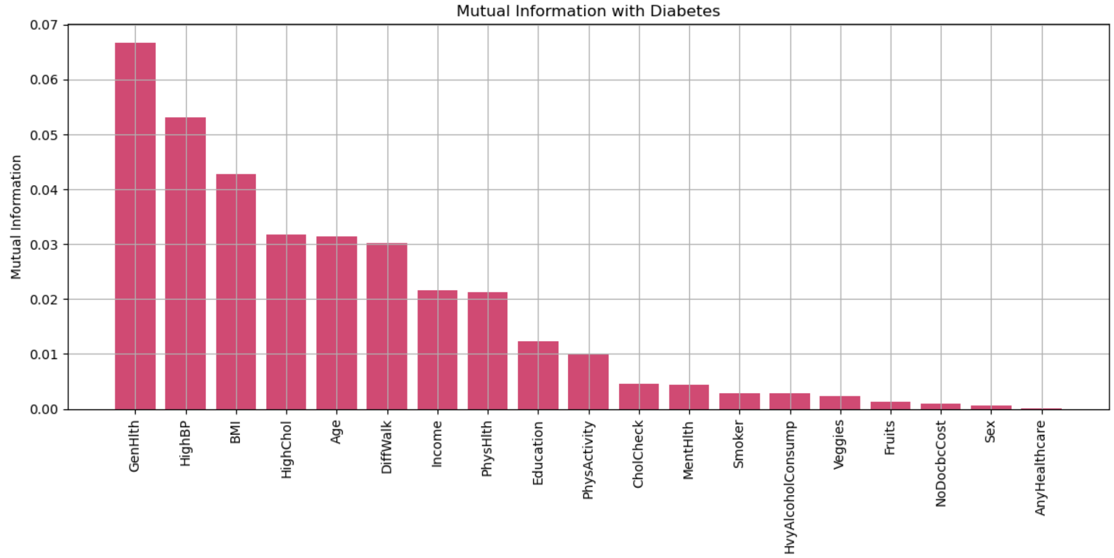
Overall mutual information

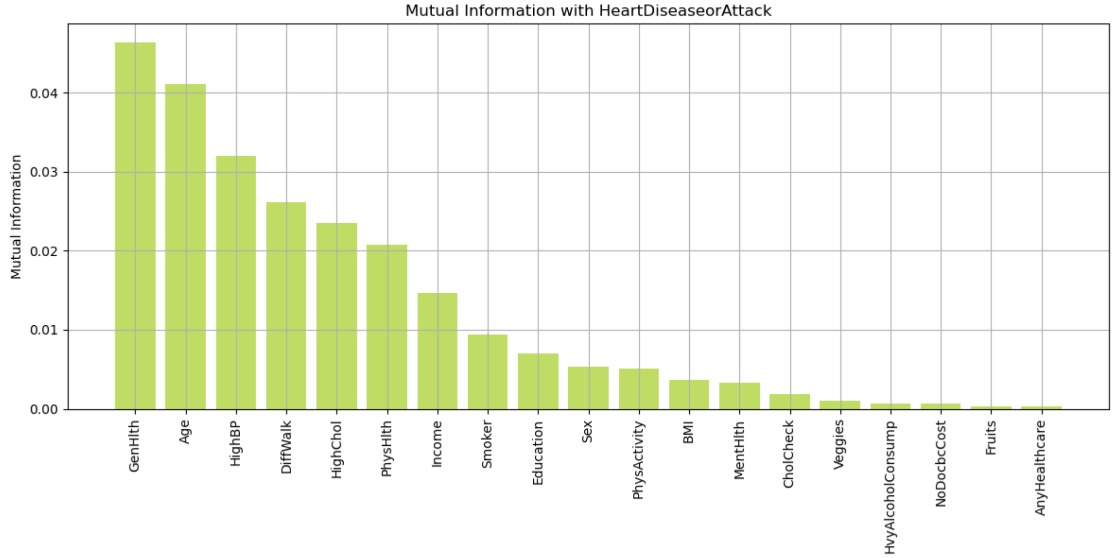
[ ]:

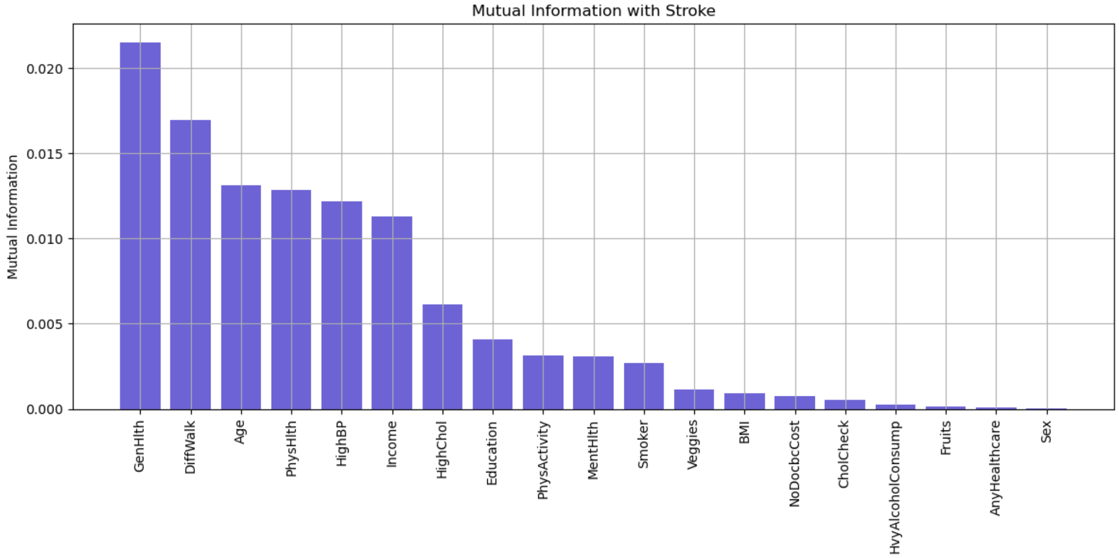
```python
def mutual_information_analysis(df, response_col):
    """Calculate mutual information for each predictor against a given response.
 ↪"""
    mi_scores = []
    response = df[response_col].values
    for column in df.columns.drop(response_col):
        mi = calculate_mi(df[column].values, response)
        mi_scores.append({'Predictor': column, 'MI': mi, 'Response':␣
 ↪response_col})
    return pd.DataFrame(mi_scores)

def evaluate_all_responses(df, responses):
    """Evaluate mutual information for all specified responses."""
    results = pd.DataFrame()
    for response in responses:
        mi_df = mutual_information_analysis(df, response)
        results = pd.concat([results, mi_df])
    return results

responses = ['HeartDiseaseorAttack', 'Stroke', 'Diabetes']
all_mi_scores = evaluate_all_responses(data, responses)
```

[ ]:

```
exclude_responses = ['HeartDiseaseorAttack', 'Stroke', 'Diabetes']


all_mi_scores_filtered = all_mi_scores[~all_mi_scores['Predictor'].
 ↪isin(exclude_responses)]


grouped = all_mi_scores_filtered.groupby('Response')


responses = all_mi_scores_filtered['Response'].unique()

fig, axes = plt.subplots(nrows=len(responses), figsize=(12, 18), sharex=False)

if len(responses) == 1:
    axes = [axes]

for (key, group), ax in zip(grouped, axes):
    mi_df = group.sort_values(by='MI', ascending=False)
    ax.bar(mi_df['Predictor'], mi_df['MI'], color=np.random.rand(3,))
    ax.set_title(f'Mutual Information with {key}')
    ax.set_ylabel('Mutual Information')
    ax.set_xticks(range(len(mi_df['Predictor'])))
    ax.set_xticklabels(mi_df['Predictor'], rotation=90)
    ax.grid(True)

fig.tight_layout()
plt.show()
```


Mutual Information with Diabetes
Mutual Information
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
GenHlth
HighBP
BMI
HighChol
Age
DiffWalk
Income
PhysHlth
Education
PhysActivity
CholCheck
MentHlth
Smoker
HvyAlcoholConsump
Veggies
Fruits
NoDocbcCost
Sex
AnyHealthcare


Mutual Information with HeartDiseaseorAttack
Mutual Information
0.04
0.03
0.02
0.01
0.00
GenHlth
Age
HighBP
DiffWalk
HighChol
PhysHlth
Income
Smoker
Education
Sex
PhysActivity
BMI
MentHlth
CholCheck
Veggies
HvyAlcoholConsump
NoDocbcCost
Fruits
AnyHealthcare


Mutual Information with Stroke
Mutual Information
0.020
0.015
0.010
0.005
0.000
GenHlth
DiffWalk
Age
PhysHlth
HighBP
Income
HighChol
Education
PhysActivity
MentHlth
Smoker
Veggies
BMI
NoDocbcCost
CholCheck
HvyAlcoholConsump
Fruits
AnyHealthcare
Sex

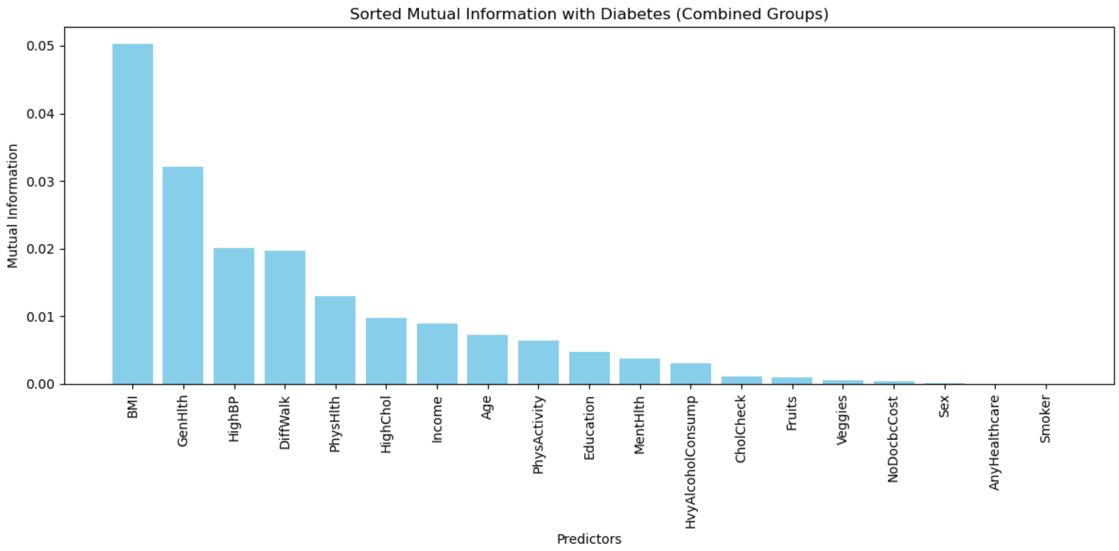
Diabetes:

[ ]:

```
data['HeartDiseaseorAttack'] = data['HeartDiseaseorAttack'].astype(int)
data['Stroke'] = data['Stroke'].astype(int)
data['Diabetes'] = data['Diabetes'].astype(int)

group_101 = data[(data['Stroke'] == 1) & (data['HeartDiseaseorAttack'] == 0) &␣
 ↪(data['Diabetes'] == 1)]
group_011 = data[(data['Stroke'] == 0) & (data['HeartDiseaseorAttack'] == 1) &␣
 ↪(data['Diabetes'] == 1)]
group_010 = data[(data['Stroke'] == 0) & (data['HeartDiseaseorAttack'] == 1) &␣
 ↪(data['Diabetes'] == 0)]
group_100 = data[(data['Stroke'] == 1) & (data['HeartDiseaseorAttack'] == 0) &␣
 ↪(data['Diabetes'] == 0)]
combined_data = pd.concat([group_101, group_011, group_010, group_100])
```

[ ]:

```
def mutual_information_analysis(df, response_col):
    mi_scores = []
    response = df[response_col].values
    for column in df.columns.drop([response_col]):
        mi = calculate_mi(df[column].values, response)
        mi_scores.append({'Predictor': column, 'MI': mi})
    return pd.DataFrame(mi_scores)

mi_df = mutual_information_analysis(combined_data, 'Diabetes')
mi_df = mi_df[~mi_df['Predictor'].isin(['HeartDiseaseorAttack', 'Stroke'])]

mi_df_sorted = mi_df.sort_values(by='MI', ascending=False)


plt.figure(figsize=(12, 6))
plt.bar(mi_df_sorted['Predictor'], mi_df_sorted['MI'], color='skyblue')
plt.title('Sorted Mutual Information with Diabetes (Combined Groups)')
plt.xlabel('Predictors')
plt.ylabel('Mutual Information')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



```
[ ]: mi_df
```

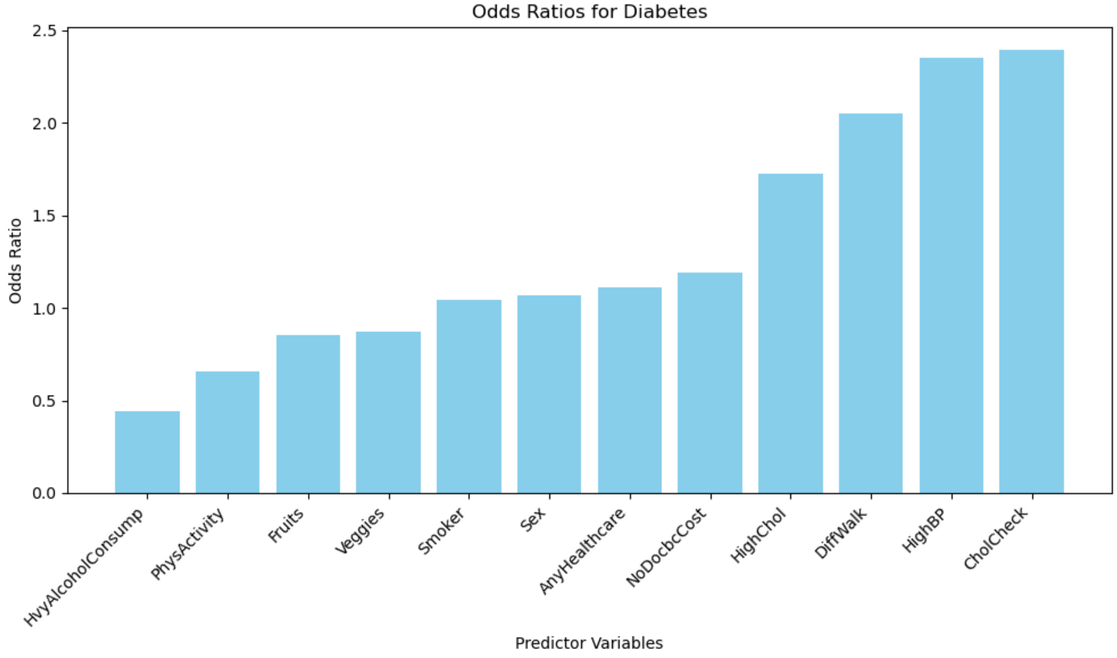
```
[ ]:             Predictor        MI
     1              HighBP  0.020153
     2            HighChol  0.009838
     3           CholCheck  0.001131
     4                 BMI  0.050230
     5              Smoker  0.000061
     7        PhysActivity  0.006416
     8              Fruits  0.000933
     9             Veggies  0.000528
     10  HvyAlcoholConsump  0.003081
     11      AnyHealthcare  0.000061
     12        NoDocbcCost  0.000472
     13            GenHlth  0.032145
     14           MentHlth  0.003759
     15           PhysHlth  0.012949
     16           DiffWalk  0.019704
     17                Sex  0.000181
     18                Age  0.007233
     19          Education  0.004747
     20             Income  0.008976
```

```
[ ]: response_vars = ["Diabetes"]
     calculate_odds_ratios(combined_data, response_vars, predictor_vars)
```

```
[ ]:            HighBP  HighChol  CholCheck    Smoker  PhysActivity    Fruits
     Veggies  HvyAlcoholConsump  AnyHealthcare  NoDocbcCost  DiffWalk       Sex
     Diabetes  2.351923  1.728502    2.39652  1.040828      0.658716  0.855143
```

```
 0.873889            0.440839       1.112381     1.193119  2.050183  1.070114
```

```
[ ]: process_data_and_plot(combined_data, response_vars, predictor_vars)
```



2 1. Common predictive factors:

The top features of diabetes in both the overall population and the subpopulations are BMI, GenHlth, and HighBP. The consistency across groups suggests that these factors arte robust features of diabetes risk, irrespective of other cardiovascular conditions.

3 2. Increased Importance of BMI:

The mutual information for BMI is the subpopulations has been siginificantly increased compared to the overall dataset, indicating that BMI plays a more important role in diabetes risk among individuals who are already suffering from either stroke or heart disease.

4 3. Independent Implications:

Although there are some changes in rank for features, a similar pattern of the mutual information between the population(different disease conditions) and the subpopulation (get either stroke or heart disease or attack, but not both) implies that these 3 risk factors do not necessarily influence each other directly.

5 4. The effection of CholCheck on Diabetes:

The odds ratio of cholesterol tests is high when patients have diabetes, indicating that most patients with diabetes have received cholesterol tests. And their mutual information value is low, which means that doing a cholesterol test will not affect the patient's development of diabetes.

```
[ ]: combined_data['fused_covariate'] = combined_data[['Smoker', 'HighChol']].
      ↪astype(str).agg('_'.join, axis=1)
```

```
[ ]: combined_data
```

```
[ ]:         HeartDiseaseorAttack  HighBP  HighChol  CholCheck   BMI  Smoker  Stroke
Diabetes  PhysActivity  Fruits  Veggies  HvyAlcoholConsump  AnyHealthcare
NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex   Age  Education  Income
fused_covariate
30                         0     1.0       1.0        1.0  34.0     1.0       1
1           1.0     0.0      0.0                0.0            1.0          0.0
4.0       0.0       7.0       1.0  0.0   9.0        5.0     4.0         1.0_1.0
93                         0     0.0       1.0        1.0  29.0     1.0       1
1           1.0     0.0      0.0                0.0            1.0          1.0
4.0      30.0      10.0       1.0  0.0  11.0        5.0     2.0         1.0_1.0
217                        0     1.0       1.0        1.0  28.0     1.0       1
1           0.0     0.0      1.0                0.0            1.0          0.0
4.0       0.0       0.0       0.0  0.0  12.0        4.0     1.0         1.0_1.0
260                        0     0.0       1.0        1.0  27.0     0.0       1
1           1.0     1.0      1.0                0.0            1.0          0.0
2.0       0.0      14.0       0.0  0.0  13.0        4.0     5.0         0.0_1.0
275                        0     1.0       1.0        1.0  32.0     1.0       1
1           1.0     0.0      1.0                0.0            1.0          0.0
5.0       0.0      30.0       1.0  1.0   8.0        5.0     7.0         1.0_1.0
…                      …     …       …        …   …     …     …
…           …     …      …                …            …
…      …       …       …       …  …   …        …     …
…
253129                     0     0.0       0.0        0.0  34.0     1.0       1
0           1.0     1.0      1.0                0.0            1.0          0.0
4.0       0.0       2.0       1.0  0.0   3.0        4.0     2.0         1.0_0.0
253332                     0     1.0       1.0        1.0  30.0     0.0       1
0           0.0     1.0      1.0                0.0            1.0          0.0
3.0       0.0      30.0       0.0  1.0  11.0        4.0     6.0         0.0_1.0
253387                     0     0.0       0.0        1.0  33.0     1.0       1
0           1.0     0.0      1.0                0.0            1.0          0.0
2.0       0.0       0.0       0.0  1.0   5.0        6.0     7.0         1.0_0.0
253531                     0     0.0       1.0        1.0  21.0     0.0       1
0           1.0     1.0      1.0                0.0            1.0          0.0
4.0       5.0      25.0       1.0  0.0   3.0        5.0     8.0         0.0_1.0
253553                     0     0.0       0.0        1.0  25.0     0.0       1
0           1.0     1.0      1.0                0.0            1.0          0.0
```

```
2.0        0.0        0.0        0.0  0.0   4.0         6.0      5.0           0.0_0.0

[26311 rows x 23 columns]
```

[ ]:

```python
import scipy.stats


contingency_table = pd.crosstab(combined_data['Diabetes'],␣
 ↪combined_data['fused_covariate'])

def conditional_entropy(x, y):
    contingency_table = pd.crosstab(x, y)
    return scipy.stats.entropy(contingency_table.values.flatten())

cond_entropy = conditional_entropy(combined_data['Diabetes'],␣
 ↪combined_data['fused_covariate'])

print('Conditional Entropy:', cond_entropy)
```

```
Conditional Entropy: 1.9281597991522923
```

[ ]:

```python
from sklearn.metrics import mutual_info_score

def mutual_information(x, y):
    return mutual_info_score(x, y)

mi = mutual_information(combined_data['Diabetes'],␣
 ↪combined_data['fused_covariate'])

print('Mutual Information:', mi)
```

```
Mutual Information: 0.006862619539790241
```

Heart Disease

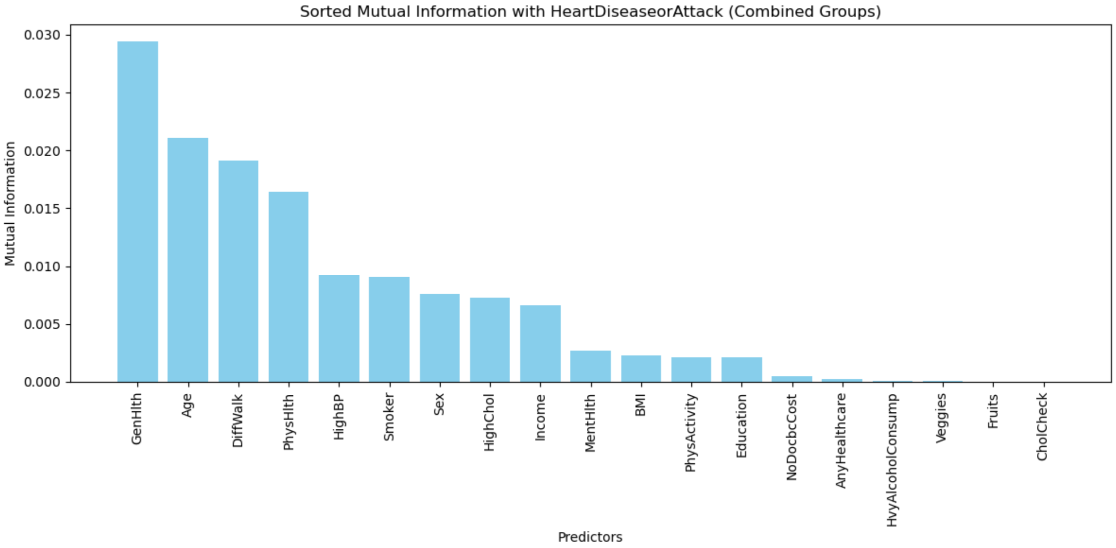
[ ]:

```python
group_100 = data[(data['Stroke'] == 1) & (data['HeartDiseaseorAttack'] == 0) &␣
 ↪(data['Diabetes'] == 0)]
group_001 = data[(data['Stroke'] == 0) & (data['HeartDiseaseorAttack'] == 0) &␣
 ↪(data['Diabetes'] == 1)]
group_110 = data[(data['Stroke'] == 1) & (data['HeartDiseaseorAttack'] == 1) &␣
 ↪(data['Diabetes'] == 0)]
group_011 = data[(data['Stroke'] == 0) & (data['HeartDiseaseorAttack'] == 1) &␣
 ↪(data['Diabetes'] == 1)]
combined_data = pd.concat([group_100, group_001, group_110, group_011])
```

[ ]:

```python
mi_df = mutual_information_analysis(combined_data, 'HeartDiseaseorAttack')
mi_df = mi_df[~mi_df['Predictor'].isin(['Diabetes', 'Stroke'])]
```

```
mi_df_sorted = mi_df.sort_values(by='MI', ascending=False)


plt.figure(figsize=(12, 6))
plt.bar(mi_df_sorted['Predictor'], mi_df_sorted['MI'], color='skyblue')
plt.title('Sorted Mutual Information with HeartDiseaseorAttack (Combined␣
  ↪Groups)')
plt.xlabel('Predictors')
plt.ylabel('Mutual Information')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



```
[ ]: mi_df_sorted
```

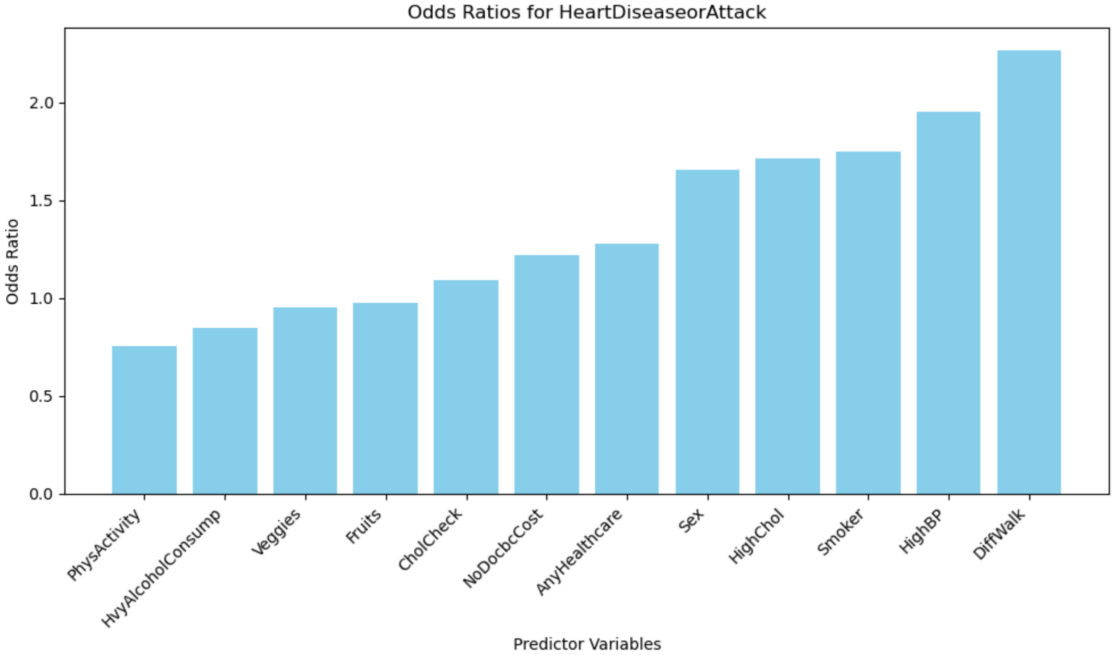
```
[ ]:           Predictor        MI
     13          GenHlth  0.029390
     18              Age  0.021042
     16         DiffWalk  0.019131
     15         PhysHlth  0.016411
     0            HighBP  0.009252
     4            Smoker  0.009072
     17              Sex  0.007560
     1          HighChol  0.007294
     20           Income  0.006593
     14         MentHlth  0.002690
     3               BMI  0.002250
     7      PhysActivity  0.002149
```

```
19          Education  0.002132
12         NoDocbcCost  0.000489
11       AnyHealthcare  0.000264
10  HvyAlcoholConsump  0.000091
9             Veggies  0.000051
8              Fruits  0.000017
2           CholCheck  0.000009
```

```
[ ]: response_vars = ["HeartDiseaseorAttack"]
     calculate_odds_ratios(combined_data, response_vars, predictor_vars)
```

```
[ ]:                           HighBP  HighChol  CholCheck   Smoker  PhysActivity
     Fruits   Veggies  HvyAlcoholConsump  AnyHealthcare  NoDocbcCost  DiffWalk
     Sex
     HeartDiseaseorAttack  1.949897  1.713856   1.094686  1.75172       0.75795
     0.975764  0.952792          0.846939       1.280217     1.221958  2.267512
     1.658707
```

```
[ ]: process_data_and_plot(combined_data, response_vars, predictor_vars)
```



covariate between HighBP and….

$I((X_1, X_2), Y)$

```
[ ]: from sklearn.metrics import mutual_info_score
```

```
predictor_vars = ['HighChol', 'CholCheck', 'Smoker', 'PhysActivity', 'Fruits',␣
 ↪'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare', 'NoDocbcCost', 'GenHlth',␣
 ↪'MentHlth', 'PhysHlth', 'DiffWalk', 'Sex', 'Age', 'Education', 'Income']

combined_data['HighBP'] = combined_data['HighBP'].astype(str)


for predictor in predictor_vars:
    combined_data[predictor] = combined_data[predictor].astype(str)
    covariate_name = f'Covariate_HighBP_{predictor}'
    combined_data[covariate_name] = combined_data['HighBP'] + '_' +␣
 ↪combined_data[predictor]
    combined_data[covariate_name] = combined_data[covariate_name].
 ↪astype('category')


combined_data['HeartDiseaseorAttack'] = combined_data['HeartDiseaseorAttack'].
 ↪astype('category')
```

eco

[ ]:

```
def ecological_terms(df, response_col):
    eco_scores = []

    if not pd.api.types.is_categorical_dtype(df[response_col]):
        df[response_col] = df[response_col].astype('category')

    response = df[response_col].cat.codes

    for column in df.columns.drop(response_col):
        try:
            parts = column.split('_')
            X1 = parts[1]
            X2 = parts[2]

            # Ensure columns are converted to 'category' dtype
            for col in [column, X1, X2]:
                if not pd.api.types.is_categorical_dtype(df[col]):
                    df[col] = df[col].astype('category')

            I_X1_X2_Y = mutual_info_score(df[column].cat.codes, response)
            I_X1_Y = mutual_info_score(df[X1].cat.codes, response)
            I_X2_Y = mutual_info_score(df[X2].cat.codes, response)
            I_X1_X2 = mutual_info_score(df[X1].cat.codes, df[X2].cat.codes)

            eco = I_X1_X2_Y - I_X1_Y - I_X2_Y + I_X1_X2
            eco_scores.append({'Predictor': column, 'Eco': eco, 'I_X1_X2':␣
↪I_X1_X2})
        except Exception as e:

            eco_scores.append({'Predictor': column, 'Eco': np.nan, 'I_X1_X2':␣
↪np.nan})

    return pd.DataFrame(eco_scores)
```

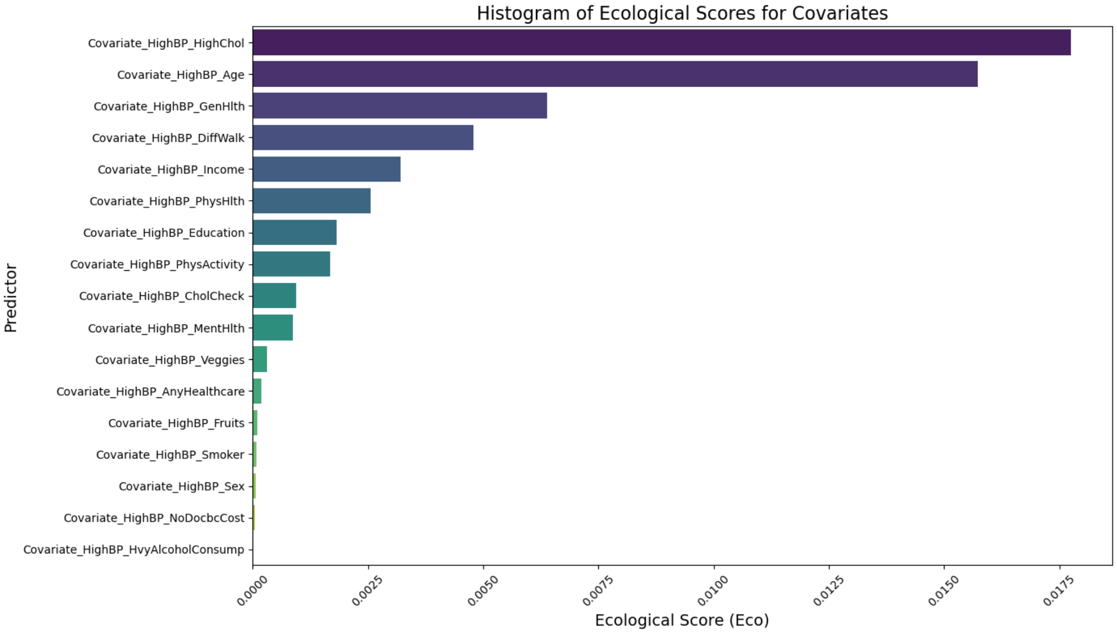
[ ]:
```
eco_covariates = ecological_terms(combined_data, 'HeartDiseaseorAttack')
eco_covariates_filtered = eco_covariates[eco_covariates['Predictor'].str.
↪startswith('Covariate_HighBP')]
```

[ ]:
```
print(eco_covariates_filtered)
```

```
                             Predictor       Eco       I_X1_X2
21           Covariate_HighBP_HighChol  0.017759  1.937099e-02
22          Covariate_HighBP_CholCheck  0.000936  8.920603e-04
23             Covariate_HighBP_Smoker  0.000087  2.621215e-04
24       Covariate_HighBP_PhysActivity  0.001686  1.947050e-03
25             Covariate_HighBP_Fruits  0.000099  1.053861e-04
26            Covariate_HighBP_Veggies  0.000301  3.170649e-04
27  Covariate_HighBP_HvyAlcoholConsump  0.000002  7.523681e-10
28      Covariate_HighBP_AnyHealthcare  0.000187  2.206925e-04
29        Covariate_HighBP_NoDocbcCost  0.000054  3.415255e-06
30            Covariate_HighBP_GenHlth  0.006385  8.380672e-03
31           Covariate_HighBP_MentHlth  0.000873  6.307322e-04
32           Covariate_HighBP_PhysHlth  0.002559  3.177737e-03
33           Covariate_HighBP_DiffWalk  0.004788  6.487578e-03
34                Covariate_HighBP_Sex  0.000060  1.325738e-05
35                Covariate_HighBP_Age  0.015731  1.767595e-02
36          Covariate_HighBP_Education  0.001813  2.107878e-03
37             Covariate_HighBP_Income  0.003208  3.935339e-03
```

[ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

eco_covariates_filtered_sorted = eco_covariates_filtered.sort_values(by='Eco',␣
↪ascending=False)

plt.figure(figsize=(14, 8))
sns.barplot(
    x='Eco',
    y='Predictor',
    data=eco_covariates_filtered_sorted,
    palette='viridis'
)
```

```
plt.xlabel('Ecological Score (Eco)', fontsize=14)
plt.ylabel('Predictor', fontsize=14)
plt.title('Histogram of Ecological Scores for Covariates', fontsize=16)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



5.0.1 1. Common predictive factors:

Common high mutual information factors: GenHlth, Age, DiffWalk.

5.0.2 2. Unusual Features:

HighBP: It was high in overall population, but decrease in the subpopulation and relatively low. Intitutively, high BP should have postive association with heart disease or attack.

5.0.3 3. Study of Covariate between HighBP and HighChol:(based on Unusual Features)

First, the individual MI values for high cholesterol and high blood pressure were in the middle of the overall MI values in response to heart disease. However, when hypertension and high cholesterol were used as covariate variables, ecological scores higher than individual MI values indicated a strong interaction effect. The presence of one of these conditions (high cholesterol) amplifies the effect of the other (high blood pressure) on heart disease risk. So when high blood pressure is combined with high cholesterol, the risk of heart disease increases significantly.

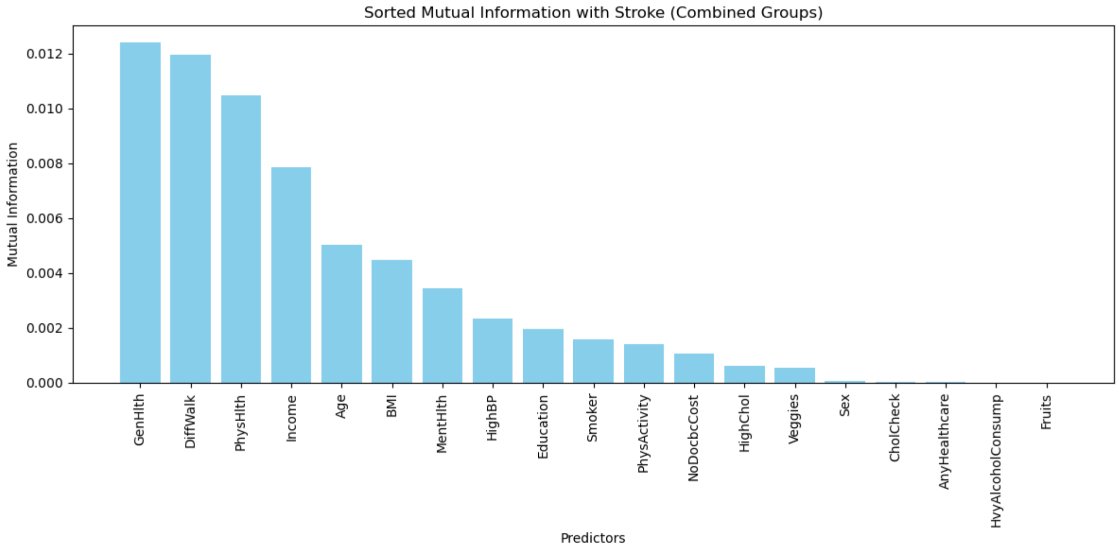
Stroke

[ ]:

```
group_110 = data[(data['Stroke'] == 1) & (data['HeartDiseaseorAttack'] == 1) &␣
  ↪(data['Diabetes'] == 0)]
group_101 = data[(data['Stroke'] == 1) & (data['HeartDiseaseorAttack'] == 0) &␣
  ↪(data['Diabetes'] == 1)]
group_010 = data[(data['Stroke'] == 0) & (data['HeartDiseaseorAttack'] == 1) &␣
  ↪(data['Diabetes'] == 0)]
group_001 = data[(data['Stroke'] == 0) & (data['HeartDiseaseorAttack'] == 0) &␣
  ↪(data['Diabetes'] == 1)]
combined_data = pd.concat([group_110, group_101, group_010, group_001])
```

[ ]:

```
mi_df = mutual_information_analysis(combined_data, 'Stroke')
mi_df = mi_df[~mi_df['Predictor'].isin(['Diabetes', 'HeartDiseaseorAttack'])]

mi_df_sorted = mi_df.sort_values(by='MI', ascending=False)

# Plotting
plt.figure(figsize=(12, 6))
plt.bar(mi_df_sorted['Predictor'], mi_df_sorted['MI'], color='skyblue')
plt.title('Sorted Mutual Information with Stroke (Combined Groups)')
plt.xlabel('Predictors')
plt.ylabel('Mutual Information')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



[ ]:

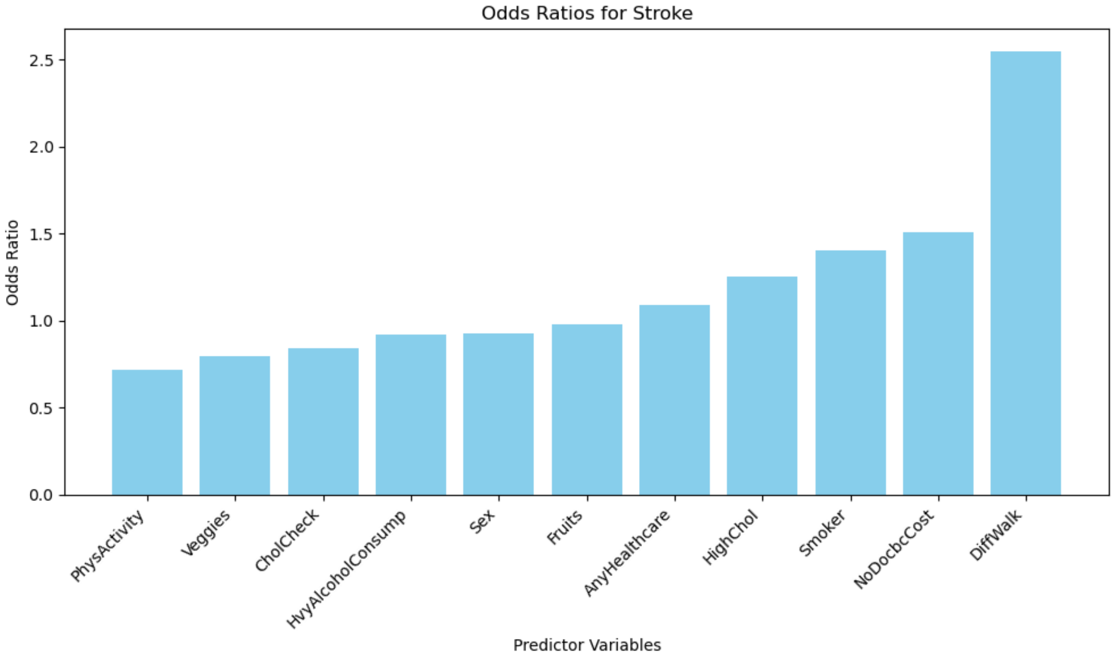
```
mi_df_sorted
```

```
[ ]:            Predictor        MI
    13            GenHlth  0.012391
    16           DiffWalk  0.011969
    15           PhysHlth  0.010487
    20             Income  0.007849
    18                Age  0.005036
    4                 BMI  0.004468
    14           MentHlth  0.003446
    1              HighBP  0.002333
    19          Education  0.001946
    5              Smoker  0.001594
    7        PhysActivity  0.001424
    12        NoDocbcCost  0.001054
    2            HighChol  0.000628
    9             Veggies  0.000563
    17                Sex  0.000080
    3           CholCheck  0.000017
    11      AnyHealthcare  0.000017
    10  HvyAlcoholConsump  0.000012
    8              Fruits  0.000005
```

```
[ ]: response_vars = ["Stroke"]
     calculate_odds_ratios(combined_data, response_vars, predictor_vars)
```

```
[ ]:         HighChol  CholCheck    Smoker  PhysActivity    Fruits   Veggies
    HvyAlcoholConsump  AnyHealthcare  NoDocbcCost  GenHlth  MentHlth  PhysHlth
    DiffWalk       Sex  Age  Education  Income
    Stroke  1.252869   0.843135  1.405768      0.720639  0.980542  0.795718
    0.91882       1.091736     1.507446      NaN       NaN       NaN  2.548613
    0.927313  NaN        NaN     NaN
```

```
[ ]: process_data_and_plot(combined_data, response_vars, predictor_vars)
```



5.0.4 1. Common predictive factors:

Common high mutual information factors: GenHlth, DiffWalk, PhysHlth. ### 2. Although the ranks of the features varies, but in general, the ranks are not too far away to its orignial position.

6 Overall:

The similar patterns of the rank of the mutual information indicates that the impact of the features are consisitent whether the other conditions are present or not.

Similar patterns of mutual information across these subpopulations indicate that certain predictors are robustly associated with a carnodisease irrespective of the presence of one of the other two diseases. For instance, if BMI consistently shows high MI with diabetes regardless of the presence or absence of stroke or heart disease, it suggests that interventions targeting BMI are likely to be effective in managing diabetes risk universally.

GenHealth's consistent ranking as a top predictor in both the overall population and subpopulations suggests it holds a central role in predicting the risk of these major health conditions. This implies a strong association between overall perceived health status and the risk of developing stroke, diabetes, and heart disease.