University Name: the State University of New York at Buffalo

Course Name: CSE 587 Data Intensive Computing

Title: Developing a Predictive Model for Heart Disease Risk Assessment Using General Health Survey Data

Group Members:

- Stuent1 id: 50560458
- Stuent1 name: Xiaofeng Chen
- Stuent2 id: 50560532
- Stuent2 name: Li Lu
- Stuent3 id: 50565976
- Stuent3 name: Xueyi Yang

Supervisor: Dr. Shamsad Parvin

Date: February 26, 2024

Problem Statement

Title: Developing a Predictive Model for Heart Disease Risk Assessment Using General Health Survey Data

Problem Statement:
Heart disease remains the leading cause of mortality globally. And it is now killing more people than ever before. However, not everyone goes to the hospital for professional heart examinations because of some inconspicuous symptoms. Thus, many people miss the best opportunity for preventive treatment, leading to heart attack and even death. Based on this common phenomenon, this project aims to address this critical issue by leveraging reliable data sources (such as general health surveys, rather than specialized cardiac examination data), to establish a predictive model for assessing the risk of heart attack among the general population in the U.S.

Background and Significance:
The prevalence of heart disease has continued to rise over the past two decades, making it a significant public health concern worldwide. According to reports from The World Health Organization (WHO)[1], heart disease claims more lives than any other cause, underscoring the urgent need for effective preventative measures. However, a considerable number of individuals do not seek medical attention until they experience severe symptoms or complications, leading to delayed diagnosis and treatment. This delay significantly reduces the effectiveness of interventions and increases the risk of adverse outcomes, including heart attacks and death.

Potential Contribution of the Project:
This project holds the potential to make a substantial contribution to the field of public health by providing a proactive approach to heart disease prevention. By developing a predictive model based on general health survey data, healthcare professionals can identify individuals at high risk of heart disease before the onset of symptoms. Moreover, the model can empower individuals to self-assess their risk of heart disease without going to hospitals for professional check, enabling them to make informed decisions about their health and lifestyle choices. This proactive approach not only facilitates early intervention but also promotes personal responsibility for health management, ultimately reducing the incidence of heart disease and its associated complications. Therefore, the project's contribution extends beyond clinical practice to empower individuals in managing their health effectively.

Data Source

This dataset source comes from Kaggle and is said to be originally from CDC, with link as below:

https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease

The dataset is a part of Behavior Risk Factor Surveillance System(BRFSS), which conducts annual telephone surveys to collect data on the health status of U.S. residents. Established in 1984 with 15 states, BRFSS now collects data in all 50 states, as well as the District of Columbia, and three U.S. territories. BRFSS completes more than 400,000 adult interviews each year, making it the largest and longest-running health survey system in the world. We choose the dataset which is published in 2022.

The raw dataset contains 40 features and more than 400,000 rows with many missing values. The meaning of features is listed as below:

Features	**Meaning**
State	the U.S. state where the individual resides
Sex	gender of the individual (Male or Female)
GeneralHealth	self-reported general health status of the individual
PhysicalHealthDays	the number of physical unhealthy days in the past 30 days
MentalHealthDays	the number of mental unhealthy days in the past 30 days
LastCheckupTime	the last health examination time
PhysicalActivities	do physical exercises or not
SleepHours	average number of hours of sleep per night
RemovedTeeth	the number of teeth has been removed
HadHeartAttack	has had a heart attack or not
HadAngina	has had angina or not
HadStroke	has had a stroke or not
HadAsthma	has had asthma or not
HadSkinCancer	has had skin cancer or not
HadCOPD	has had Chronic Obstructive Pulmonary Disease (COPD) or not
HadDepressiveDisorder	has had a depressive disorder or not
HadKidneyDisease	has had kidney disease or not
HadArthritis	has had arthritis or not

HadDiabetes	has had diabetes or not
DeafOrHardOfHearing	is deaf or hard of hearing or not
BlindOrVisionDifficulty	has blindness or vision difficulty or not
DifficultyConcentrating	has difficulty in concentrating or not
DifficultyWalking	has difficulty in walking or not
DifficultyDressingBathing	has difficulty in dressing or bathing or not
DifficultyErrands	has difficulty in running errands or not
SmokerStatus	smoking status
ECigaretteUsage	e-cigarettes using status
ChestScan	has had a chest scan or not
RaceEthnicityCategory	categorized race and ethnicity
AgeCategory	age grouping
HeightInMeters	height in meters
WeightInKilograms	weight in kilograms
BMI	Body Mass Index calculated from weight/height^2
AlcoholDrinkers	drink alcohol or not
HIVTesting	has HIV testing or not
FluVaxLast12	has received a flu vaccine in the last 12 months or not
PneumoVaxEver	has ever received a pneumonia vaccine or not
TetanusLast10Tdap	time since the last tetanus vaccination within the recent 10 years
HighRiskLastYear	has been estimated at high risk for the past year
CovidPos	has COVID-19 testing positive or negative

Data Cleaning/Processing

Step 1: Remove duplicate rows

There are several duplicated rows. In the first step of data cleaning, we should remove the duplicated rows.

Step 2: Drop unwanted column

The decision to drop certain features during data cleaning usually stems from an assumption or analysis suggesting that these variables are not significantly correlated with the heart attack rate and prediction. (For example, RemovedTeeth: Dental health may not directly influence heart health, although there are some debates about the link between oral health and heart disease; LastCheckupTime: The time of the last checkup may not necessarily reflect current heart health status.

```
#Data cleaning step : Drop unwanted column
df= df.drop(columns=['RemovedTeeth', 'LastCheckupTime', 'ChestScan','DeafOrHardOfHearing','BlindOrVisionDifficulty',
                     'DifficultyConcentrating','DifficultyWalking','DifficultyDressingBathing','HadSkinCancer',
                     'DifficultyErrands','HIVTesting', 'FluVaxLast12', 'PneumoVaxEver', 'TetanusLast10Tdap',
                     'ECigaretteUsage','HighRiskLastYear',"HeightInMeters","WeightInKilograms","CovidPos","RaceEthnicityCategory"])
```

✓ 0.0s

```
df.shape
```

✓ 0.0s

```
(441915, 20)
```

Step 3: Data standardization

Specifically, we standardize the categories for the 'SmokerStatus' column. This step simplifies the categories by condensing them to more general terms, which makes them more accessible for further analysis.

```
# Data cleaning step: Data standardization
df['SmokerStatus'].replace({'Never smoked' : 'Never'}, inplace=True)
df['SmokerStatus'].replace({'Former smoker' : 'Former'}, inplace=True)
df['SmokerStatus'].replace({'Current smoker - now smokes some days' : 'Sometimes'}, inplace=True)
df['SmokerStatus'].replace({'Current smoker - now smokes every day' : 'Everyday'}, inplace=True)
df['SmokerStatus'].value_counts()
```

✓ 0.0s

```
SmokerStatus
Never        244607
Former       113004
Everyday      35682
Sometimes     13822
Name: count, dtype: int64
```

Step 4: Categorical data consolidation

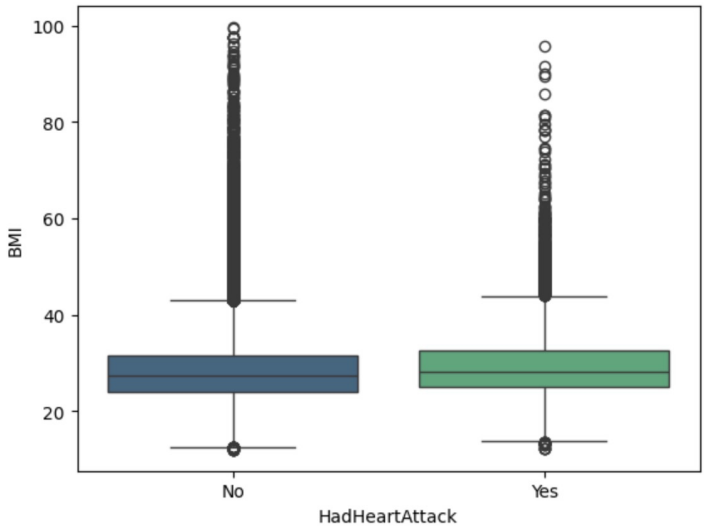
This step in data cleaning helps ensure that the data within the 'AgeCategory' column is consistent and standardized, which is necessary for accurate analysis and comparison. It also simplifies the representation of the data, which can can prove advantageous for visualization, machine learning modeling, and reporting purposes.

```
age_mapping = {
    'Age 80 or older': '80+',
    'Age 55 to 59': '55-59',
    'Age 40 to 44': '40-44',
    'Age 75 to 79': '75-79',
    'Age 70 to 74': '70-74',
    'Age 65 to 69': '65-69',
    'Age 60 to 64': '60-64',
    'Age 50 to 54': '50-54',
    'Age 45 to 49': '45-49',
    'Age 35 to 39': '35-39',
    'Age 25 to 29': '25-29',
    'Age 30 to 34': '30-34',
    'Age 18 to 24': '18-24'
}
df['AgeCategory'] = df['AgeCategory'].map(age_mapping)
df['AgeCategory'] = df['AgeCategory'].map(age_mapping).fillna(df['AgeCategory'])
df['AgeCategory'].value_counts()
```

```
AgeCategory
65-69    46793
60-64    44191
70-74    43194
55-59    36611
80+      35736
50-54    33436
75-79    32231
40-44    29778
45-49    28392
35-39    28371
18-24    26792
30-34    25687
25-29    21901
Name: count, dtype: int64
```

Step 5: Check outlier

There are many instances of excessively high BMI values. We examined whether these values could impact the likelihood of a heart attack and found that these outliers had no significant effect. Consequently, We decided to remove these rows.




```
df['HadHeartAttack'].value_counts()
```

[26] ✓ 0.0s

```
HadHeartAttack
No     416807
Yes     25108
Name: count, dtype: int64
```

```
df_t=df.copy()
df_filtered = df_t[(df_t['BMI'] <= 50) | (df_t['BMI'].isnull())]
df_filtered.shape
```

[27] ✓ 0.1s

```
(438327, 20)
```

```
df_filtered['HadHeartAttack'].value_counts()
```

[28] ✓ 0.0s

```
HadHeartAttack
No     413468
Yes     24859
Name: count, dtype: int64
```

```
df = df[(df['BMI'] <= 50) | (df['BMI'].isnull())]
df['HadHeartAttack'].value_counts()
```

[29] ✓ 0.1s

Step 6 : Missing Value Imputation

As many algorithms cannot handle missing values and may produce erroneous or biased outputs if the missing values are not properly addressed, we replace all missing "BMI" data with the mean value.

```
#Data Cleaning : Missing Value Imputation
from sklearn.impute import SimpleImputer
imputer=SimpleImputer(missing_values=np.nan,strategy="mean")
df[["BMI"]]=imputer.fit_transform(df[["BMI"]])
df["BMI"].value_counts()
```

✓ 0.0s

```
BMI
28.527883    47968
26.630000     4236
27.460000     3260
24.410000     3167
27.440000     3115
             ...
47.380000        1
59.130000        1
20.170000        1
47.280000        1
48.630000        1
Name: count, Length: 3978, dtype: int64
```

Step 7: Missing Value Imputation

We selected these three factors in combination with "BMI" as the health_factor. To address missing values, we replace them with the mode values.

```
#Data cleaning step: Missing Value Imputation
columns_to_fill = ["PhysicalHealthDays", "MentalHealthDays", "SleepHours"]
for col in columns_to_fill:
    df[col] = df[col].transform(lambda x: x.fillna(x.mode().iloc[0]))
```

✓ 0.0s

Step 8 : Drop missing values

For the remaining missing values, we just drop them. This drop operation does not change the ratio of heart attack sample in our dataset.

```
#Data cleaning step : Drop missing values
df = df.dropna(subset=['AlcoholDrinkers','AgeCategory','SmokerStatus',"HadDiabetes","HadArthritis","HadKidneyDisease","HadDepressiveDisorder",
                       "HadCOPD","HadAsthma","HadStroke","HadAngina","GeneralHealth","PhysicalActivities"] )
df.isnull().sum()
df['HadHeartAttack'].value_counts()
```

✓ 0.3s

```
HadHeartAttack
No     351437
Yes     20023
Name: count, dtype: int64
```

Step 9: Feature Engineering

This step defines a new function that takes a numerical range representing "hours of sleep" and categorizes it into a sleep duration category. This transformation creates a more meaningful categorical variable compared to the former one, which can be useful in following data analysis, visualization, and machine learning models.

```
#Data cleaning 9: Feature engineering
def categorize_sleep_hours(hours):
    if hours < 5:
        return 'Very Short'
    elif 5 <= hours <= 6:
        return 'Short'
    elif 7 <= hours <= 8:
        return 'Recommended'
    elif 9 <= hours <= 10:
        return 'Long'
    else:
        return 'Very Long'

df['SleepCategory'] = df['SleepHours'].apply(categorize_sleep_hours)
sleep_cat_order = ['Very Short', 'Short', 'Recommended', 'Long', 'Very Long']
```

✓ 0.0s

Step 10: Column Rearrange

Many statistical analysis techniques and machine learning algorithms require numerical input. Transforming categorical data into numeric format allows these methods to process and analyze the data. And we use this for following Exploratory Data Analysis.

```
#Data cleaning step: column rearrange
df['HadHeartAttackNumeric'] = df['HadHeartAttack'].map({'Yes': 1, 'No': 0})
df.describe()
```

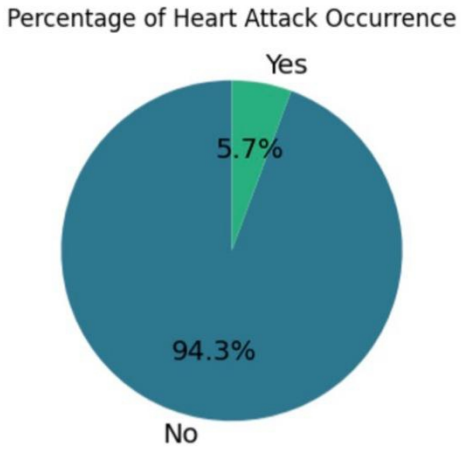
[5] ✓ 0.0s

	PhysicalHealthDays	MentalHealthDays	SleepHours	HadDepressiveDisorder	BMI	HadHeartAttackNumeric
count	371460.000000	371460.000000	371460.000000	371460.000000	371460.000000	371460.000000
mean	4.096988	4.224439	7.027010	0.208082	28.306014	0.053904
std	8.422598	8.196573	1.454334	0.405936	5.777302	0.225828
min	0.000000	0.000000	1.000000	0.000000	12.020000	0.000000
25%	0.000000	0.000000	6.000000	0.000000	24.330000	0.000000
50%	0.000000	0.000000	7.000000	0.000000	27.890000	0.000000
75%	3.000000	4.000000	8.000000	0.000000	31.250000	0.000000
max	30.000000	30.000000	24.000000	1.000000	50.000000	1.000000

Exploratory Data Analysis

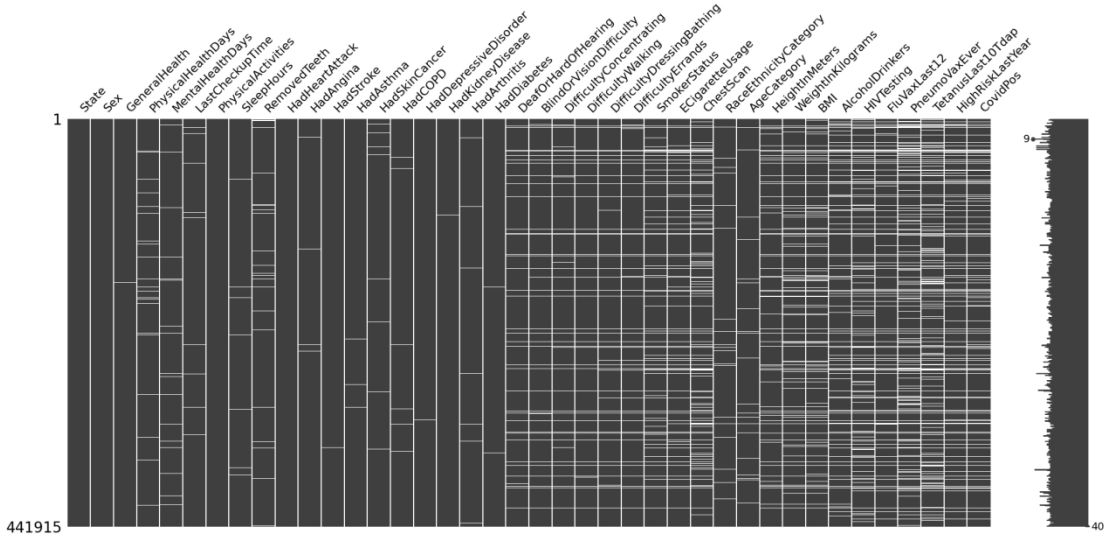
Step 1: Univariate Analysis

The process involves counting the frequency of occurrences of 'HadHeartAttack' and visualizing these frequencies to gain an understanding of the distribution across categories.



Step 2: Missing Data analysis

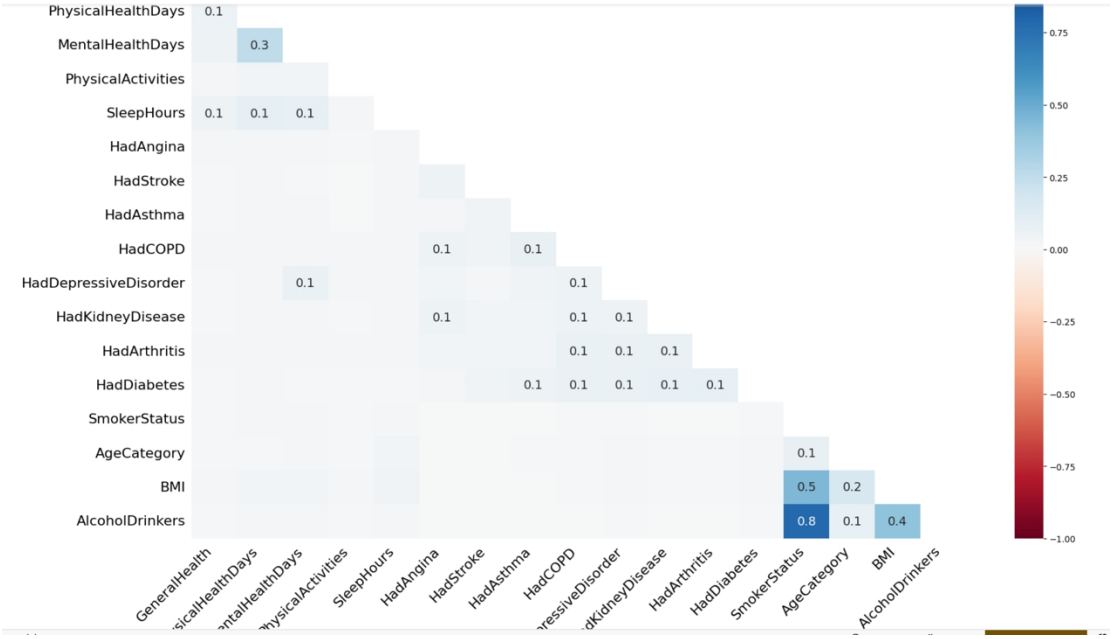
This involves identifying the presence and distribution of missing values across different variables in the dataset. Understanding where and how data is missing can inform us how to handle these missing values. Possible strategies for handling missing data include imputation, deletion, and more sophisticated methods, which may vary depending on the nature and extent of the missing data. From the graph we can see there are too much missing data in features like CovidPos.



Step 3: nullity correlation

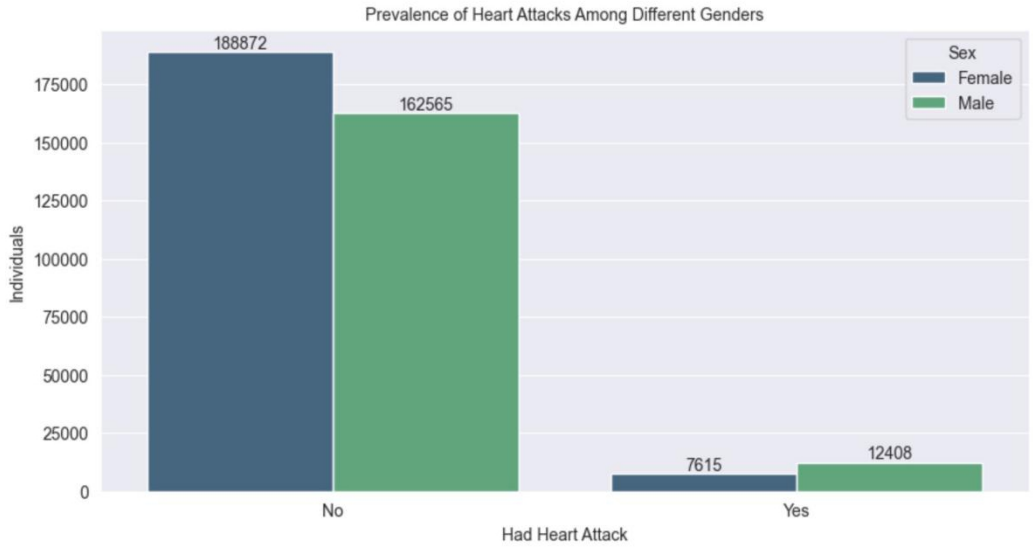
Nullity correlation between two variables ranges from -1 to 1, indicating the strength and direction of their association. A correlation of -1 implies a strong negative relationship, meaning if one variable is present, the other is likely absent. This is represented by color red in visualization. Conversely, a correlation of 1 signifies a strong positive relationship, where the presence of one variable ensures the presence of the other, depicted as blue.

For example, in the graph, we find that AlcoholDrinkers correlated positively with SmokerStatus. If AlcoholDrinkers is missing, SmokerStatus is also missing.



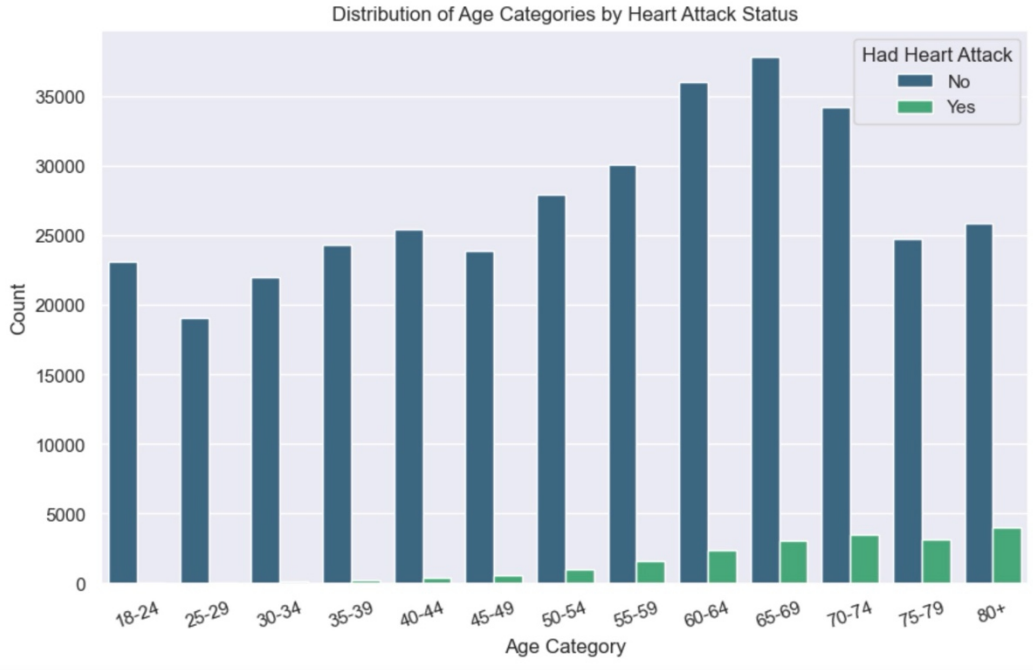
Step 4: Bivariate Analysis/Categorical Comparison

This type of analysis examines the relationship between having had a heart attack ('HadHeartAttack') and gender ('Sex'). We can see the bar for males in the 'Yes' (had a heart attack) category is higher than the bar for females.



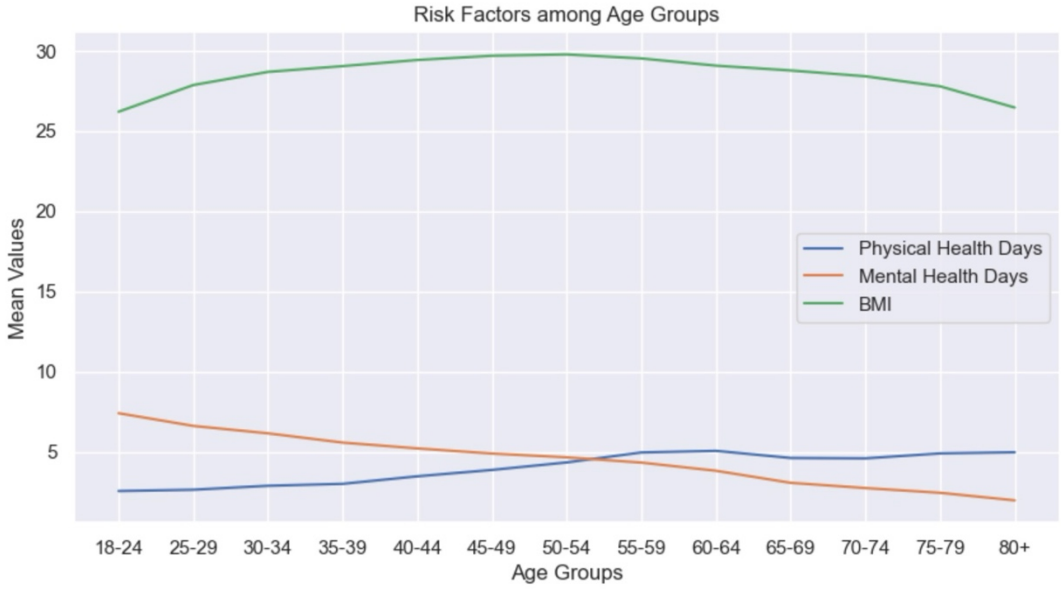
Step 5: Distribution Analysis

This specific analysis is using a count plot to show the distribution of heart attack occurrences across different age categories. Heart attacks are relatively rare among younger age groups. There is a noticeable increase in heart attack occurrences among older age groups, starting from 60-64 and becoming more pronounced from 70-74 onwards. This suggests that the likelihood of having a heart attack increases with age. The age group of 80+ has the highest proportion of heart attack occurrences, indicated by the green bar's size relative to the blue bar.



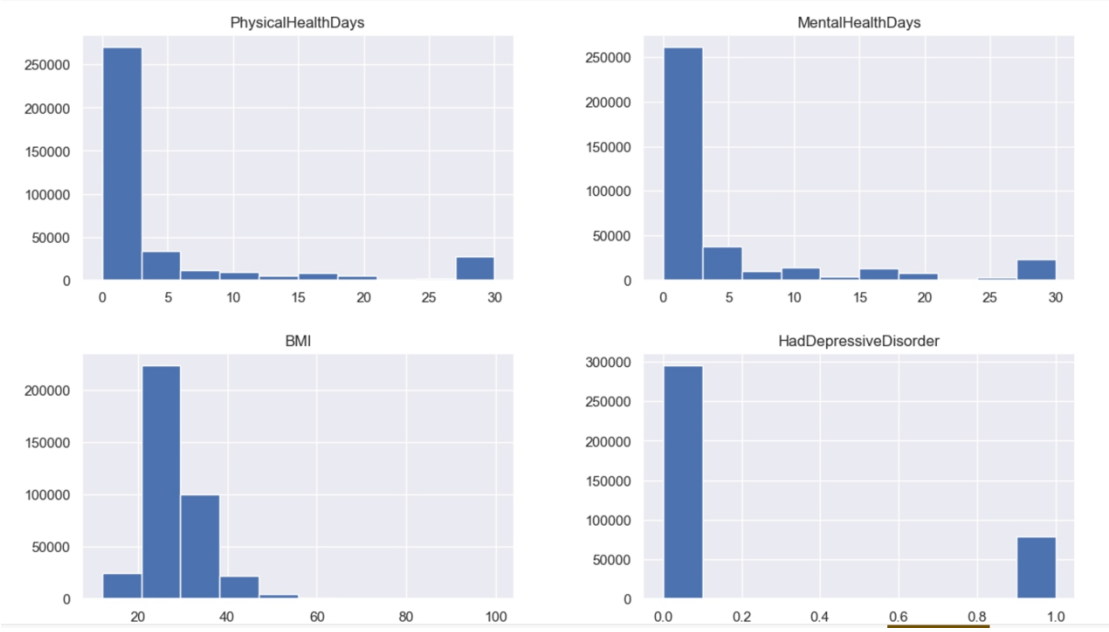
Step 6: Comparative Analysis

This step compares different age categories to examine the differences in mean values of physical health days, mental health days, and BMI. From the graph, it can be seen that the mean values of them appear to fluctuate with age, suggesting that physical health is getting worse as people grow older.



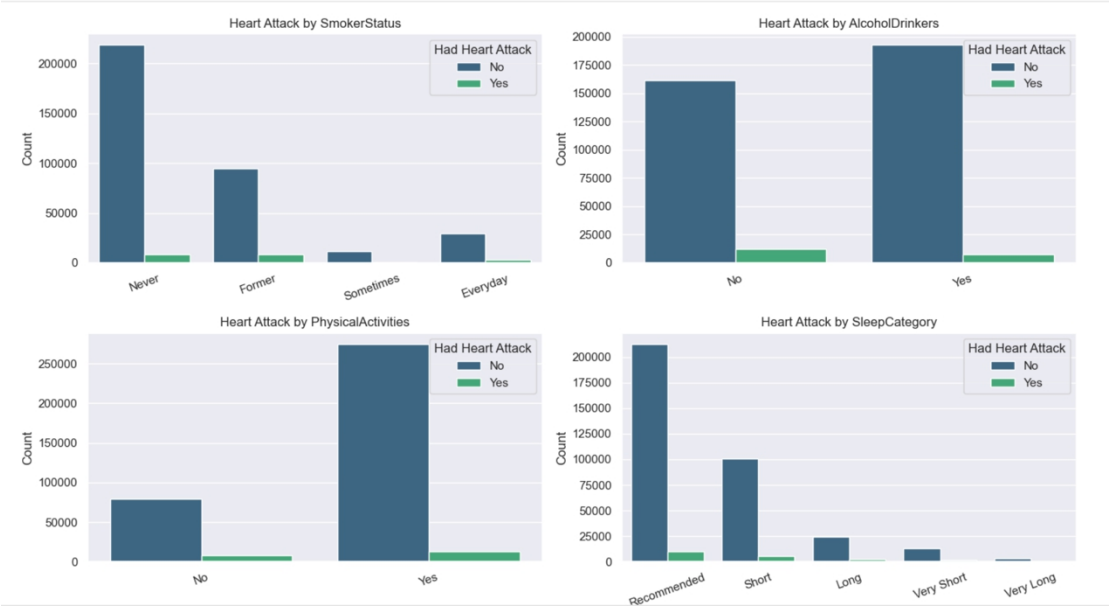
Step 7: Univariate analysis I

These four histograms illustrate the frequency distribution of individual variables, including 'PhysicalHealthDays', 'MentalHealthDays', 'BMI', and 'HadDepressiveDisorder'. These four features collectively contribute to the health_factor, as individuals are influenced by both their physical and mental health status. We will analyze this health_factor subsequently.



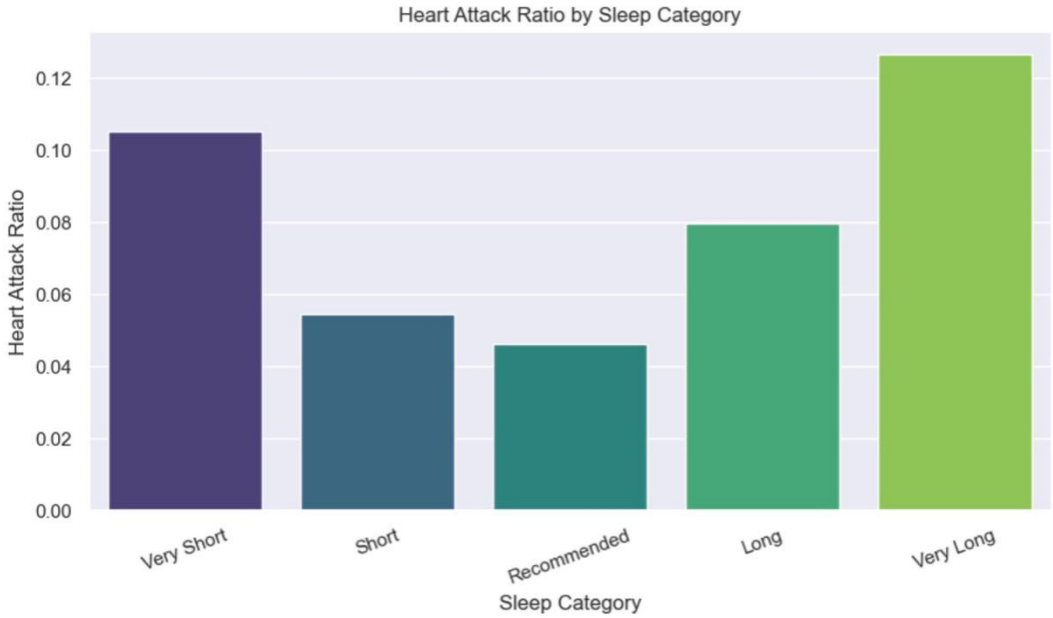
Step 8: Univariate analysis II

These four features, 'SmokerStatus', 'AlcoholDrinkers', 'PhysicalActivities', and 'SleepCategory', represent lifestyle factors. For instance, this chart likely depicts the distribution of individuals categorized as never smokers, former smokers, occasional smokers, and daily smokers, comparing the counts of those who have and have not had a heart attack. A higher proportion of individuals who have never smoked may not have had a heart attack, while there might be a higher proportion of heart attacks among former and current smokers.



Step 9: Comparative analysis

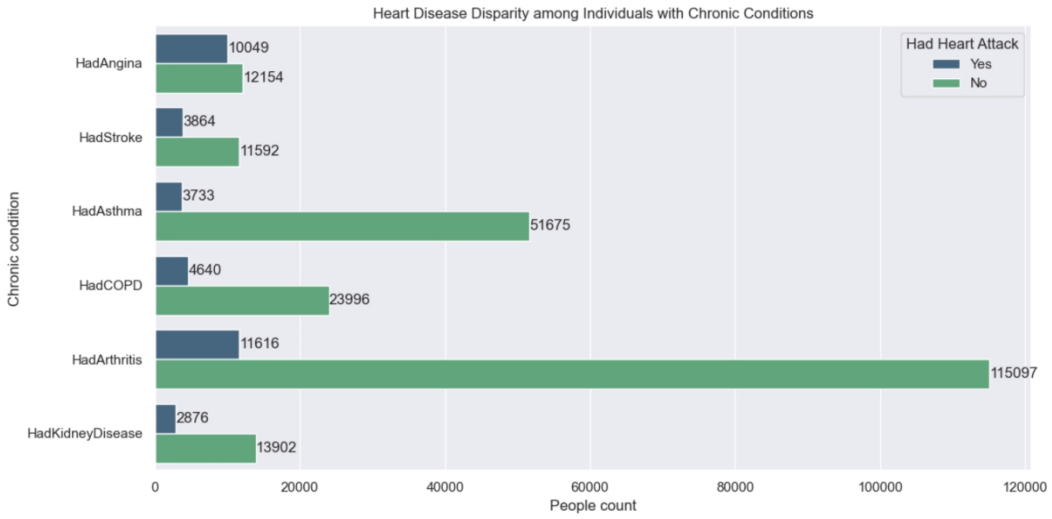
The bar plot illustrates the heart attack ratio across different sleep durations. The x-axis displays the sleep duration categories: 'Very Short', 'Short', 'Recommended', 'Long', and 'Very Long'. The y-axis represents the heart attack ratio, indicating the proportion of individuals who have experienced a heart attack within each sleep category. This chart allows for the observation of potential trends or patterns in heart attack occurrences relative to sleep duration. For example, if the bars for 'Very Short' and 'Very Long' sleep categories are higher than that of the 'Recommended' category, it may suggest that both significantly shorter and longer sleep durations are associated with a higher incidence of heart attacks compared to the recommended amount of sleep.



Step 10: Categorical analysis and comparison

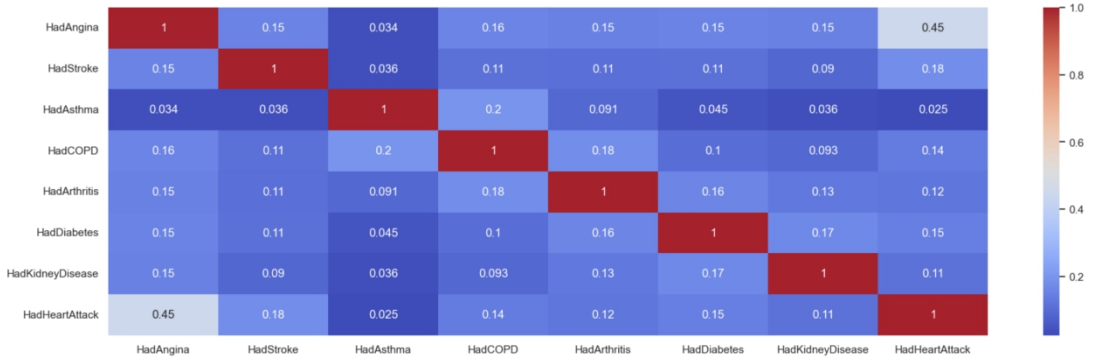
For the first graph, the green bars represent individuals with a specific chronic condition who have not had a heart attack, while the blue bars represent those who have had a heart attack. The chart visually compares the number of individuals with a particular chronic condition who have experienced a heart attack against those who have not. The relationship between categorical data (chronic conditions) and a binary outcome (Had Heart Attack: Yes or No) is analyzed.

For the second graph, it's a feature correlation analysis chart. Each value in the chart represents the probability of the co-occurrence of diseases on its x and y axes. From the last row of data, we can intuitively observe the correlation between various chronic diseases and the incidence of heart disease.




```python
chronic_diseases_numeric = chronic_diseases_corr.replace({'Yes': 1, 'No': 0})
corr_matrix = chronic_diseases_numeric.corr()
sns.heatmap(corr_matrix, annot=True,cmap='coolwarm')
plt.rcParams['figure.figsize'] = (18, 6)
plt.show()
```

Python



Summary

After EDA, there are no missing values in the dataset, and the data is well-structured, facilitating the subsequent modeling process. Additionally, the retained features are correlated with the predicted target occurrence probability. In phase 2, we will utilize the processed dataset to obtain a predictive model for heart disease occurrence under other chronic conditions, unhealthy lifestyle habits, and physical health status.

[1] https://www.who.int/news/item/09-12-2020-who-reveals-leading-causes-of-death-and-disability-worldwide-2000-2019