exploratory-data-analysis

March 18, 2024

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
import plotly.express as px
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
#reading the CVD dataset
CVD_data_path = '/Users/kalyankumar/Downloads/
↪heart_disease_health_indicators_BRFSS2015 (1).csv'
CVD_DF = pd.read_csv(CVD_data_path , low_memory=False)
```

[3]:
```
#taking look at the data
CVD_DF.head()
```

[3]:
```
   HeartDiseaseorAttack  HighBP  HighChol  CholCheck   BMI  Smoker  Stroke  \
0                   0.0     1.0       1.0        1.0  40.0     1.0     0.0
1                   0.0     0.0       0.0        0.0  25.0     1.0     0.0
2                   0.0     1.0       1.0        1.0  28.0     0.0     0.0
3                   0.0     1.0       0.0        1.0  27.0     0.0     0.0
4                   0.0     1.0       1.0        1.0  24.0     0.0     0.0

   Diabetes  PhysActivity  Fruits  …  AnyHealthcare  NoDocbcCost  GenHlth  \
0       0.0           0.0     0.0  …            1.0          0.0      5.0
1       0.0           1.0     0.0  …            0.0          1.0      3.0
2       0.0           0.0     1.0  …            1.0          1.0      5.0
3       0.0           1.0     1.0  …            1.0          0.0      2.0
4       0.0           1.0     1.0  …            1.0          0.0      2.0

   MentHlth  PhysHlth  DiffWalk  Sex   Age  Education  Income
0      18.0      15.0       1.0  0.0   9.0        4.0     3.0
1       0.0       0.0       0.0  0.0   7.0        6.0     1.0
2      30.0      30.0       1.0  0.0   9.0        4.0     8.0
3       0.0       0.0       0.0  0.0  11.0        3.0     6.0
4       3.0       0.0       0.0  0.0  11.0        5.0     4.0
```

```
[5 rows x 22 columns]
```

[4]:
```
#checking shape of dataset
CVD_DF.shape
```

[4]:
```
(253680, 22)
```

[5]:
```
# getting information of dataset
CVD_DF.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
```

[6]:
```
#checking null or missing values in dataset
null_values = CVD_DF.isna().sum()
print('\n missing values in dataset \n')
print(null_values)
```

```
 missing values in dataset
```

```
HeartDiseaseorAttack    0
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  0
Diabetes                0
PhysActivity            0
Fruits                  0
Veggies                 0
HvyAlcoholConsump       0
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64
```

[7]:
```
#number of duplicates in dataset
CVD_DF.duplicated().sum()
```

[7]:
```
23899
```

[8]:
```
#drop duplicates from dataset
CVD_DF_FINAL = CVD_DF.drop_duplicates()
```

[9]:
```
#getting information of datset after dropping duplicates
CVD_DF_FINAL.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 229781 entries, 0 to 253679
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   HeartDiseaseorAttack  229781 non-null  float64
 1   HighBP                229781 non-null  float64
 2   HighChol              229781 non-null  float64
 3   CholCheck             229781 non-null  float64
 4   BMI                   229781 non-null  float64
 5   Smoker                229781 non-null  float64
 6   Stroke                229781 non-null  float64
```

```
 7   Diabetes              229781 non-null  float64
 8   PhysActivity          229781 non-null  float64
 9   Fruits                229781 non-null  float64
 10  Veggies               229781 non-null  float64
 11  HvyAlcoholConsump     229781 non-null  float64
 12  AnyHealthcare         229781 non-null  float64
 13  NoDocbcCost           229781 non-null  float64
 14  GenHlth               229781 non-null  float64
 15  MentHlth              229781 non-null  float64
 16  PhysHlth              229781 non-null  float64
 17  DiffWalk              229781 non-null  float64
 18  Sex                   229781 non-null  float64
 19  Age                   229781 non-null  float64
 20  Education             229781 non-null  float64
 21  Income                229781 non-null  float64
dtypes: float64(22)
memory usage: 40.3 MB
```

[10]:
```
#again checking if there are any duplicates
CVD_DF_FINAL.duplicated().sum()
```

[10]: 0

[11]:
```
display(CVD_DF_FINAL)
```

```
        HeartDiseaseorAttack  HighBP  HighChol  CholCheck   BMI  Smoker  \
0                        0.0     1.0       1.0        1.0  40.0     1.0
1                        0.0     0.0       0.0        0.0  25.0     1.0
2                        0.0     1.0       1.0        1.0  28.0     0.0
3                        0.0     1.0       0.0        1.0  27.0     0.0
4                        0.0     1.0       1.0        1.0  24.0     0.0
…                          …       …         …          …     …       …
253675                   0.0     1.0       1.0        1.0  45.0     0.0
253676                   0.0     1.0       1.0        1.0  18.0     0.0
253677                   0.0     0.0       0.0        1.0  28.0     0.0
253678                   0.0     1.0       0.0        1.0  23.0     0.0
253679                   1.0     1.0       1.0        1.0  25.0     0.0

        Stroke  Diabetes  PhysActivity  Fruits  …  AnyHealthcare  \
0          0.0       0.0           0.0     0.0  …            1.0
1          0.0       0.0           1.0     0.0  …            0.0
2          0.0       0.0           0.0     1.0  …            1.0
3          0.0       0.0           1.0     1.0  …            1.0
4          0.0       0.0           1.0     1.0  …            1.0
…            …         …             …       …  …              …
253675     0.0       0.0           0.0     1.0  …            1.0
253676     0.0       2.0           0.0     0.0  …            1.0
253677     0.0       0.0           1.0     1.0  …            1.0
```

```
253678     0.0       0.0           0.0      1.0  …            1.0
253679     0.0       2.0           1.0      1.0  …            1.0

        NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex   Age  \
0               0.0      5.0      18.0      15.0       1.0  0.0   9.0
1               1.0      3.0       0.0       0.0       0.0  0.0   7.0
2               1.0      5.0      30.0      30.0       1.0  0.0   9.0
3               0.0      2.0       0.0       0.0       0.0  0.0  11.0
4               0.0      2.0       3.0       0.0       0.0  0.0  11.0
…             …      …       …       …         …  …    …
253675          0.0      3.0       0.0       5.0       0.0  1.0   5.0
253676          0.0      4.0       0.0       0.0       1.0  0.0  11.0
253677          0.0      1.0       0.0       0.0       0.0  0.0   2.0
253678          0.0      3.0       0.0       0.0       0.0  1.0   7.0
253679          0.0      2.0       0.0       0.0       0.0  0.0   9.0

        Education  Income
0             4.0     3.0
1             6.0     1.0
2             4.0     8.0
3             3.0     6.0
4             5.0     4.0
…           …      …
253675        6.0     7.0
253676        2.0     4.0
253677        5.0     2.0
253678        5.0     1.0
253679        6.0     2.0

[229781 rows x 22 columns]
```

[12]:

```python
#Data Transformation:
  # - Data normalization or scaling: Ensure that numerical features have a consistent scale.

# from the dataset HighBP, HighChol, CholCheck, Smoker, Stroke, Diabetes,
# PhysActivity columns are numerical and can be scaled
columns = ['HighBP', 'HighChol', 'CholCheck', 'Smoker', 'Stroke', 'Diabetes', 'PhysActivity']

# Apply Min-Max scaling(scale 0 to 1)
min_max_scaler = MinMaxScaler()
CVD_DF_DT = CVD_DF_FINAL.copy()
CVD_DF_DT[columns] = min_max_scaler.fit_transform(CVD_DF_DT[columns])
```

[13]:

```python
display(CVD_DF_DT)
```

```
        HeartDiseaseorAttack  HighBP  HighChol  CholCheck   BMI  Smoker  \
```

```
0                        0.0     1.0        1.0         1.0  40.0      1.0
1                        0.0     0.0        0.0         0.0  25.0      1.0
2                        0.0     1.0        1.0         1.0  28.0      0.0
3                        0.0     1.0        0.0         1.0  27.0      0.0
4                        0.0     1.0        1.0         1.0  24.0      0.0
…                          …       …          …           …    …        …
253675                   0.0     1.0        1.0         1.0  45.0      0.0
253676                   0.0     1.0        1.0         1.0  18.0      0.0
253677                   0.0     0.0        0.0         1.0  28.0      0.0
253678                   0.0     1.0        0.0         1.0  23.0      0.0
253679                   1.0     1.0        1.0         1.0  25.0      0.0

        Stroke  Diabetes  PhysActivity  Fruits  …  AnyHealthcare  \
0          0.0       0.0           0.0     0.0  …            1.0
1          0.0       0.0           1.0     0.0  …            0.0
2          0.0       0.0           0.0     1.0  …            1.0
3          0.0       0.0           1.0     1.0  …            1.0
4          0.0       0.0           1.0     1.0  …            1.0
…            …         …             …       …  …              …
253675     0.0       0.0           0.0     1.0  …            1.0
253676     0.0       1.0           0.0     0.0  …            1.0
253677     0.0       0.0           1.0     1.0  …            1.0
253678     0.0       0.0           0.0     1.0  …            1.0
253679     0.0       1.0           1.0     1.0  …            1.0

        NoDocbcCost  GenHlth  MentHlth  PhysHlth  DiffWalk  Sex   Age  \
0               0.0      5.0      18.0      15.0       1.0  0.0   9.0
1               1.0      3.0       0.0       0.0       0.0  0.0   7.0
2               1.0      5.0      30.0      30.0       1.0  0.0   9.0
3               0.0      2.0       0.0       0.0       0.0  0.0  11.0
4               0.0      2.0       3.0       0.0       0.0  0.0  11.0
…                 …        …         …         …         …    …     …
253675          0.0      3.0       0.0       5.0       0.0  1.0   5.0
253676          0.0      4.0       0.0       0.0       1.0  0.0  11.0
253677          0.0      1.0       0.0       0.0       0.0  0.0   2.0
253678          0.0      3.0       0.0       0.0       0.0  1.0   7.0
253679          0.0      2.0       0.0       0.0       0.0  0.0   9.0

        Education  Income
0             4.0     3.0
1             6.0     1.0
2             4.0     8.0
3             3.0     6.0
4             5.0     4.0
…               …       …
253675        6.0     7.0
253676        2.0     4.0
253677        5.0     2.0
```

```
253678        5.0     1.0
253679        6.0     2.0

[229781 rows x 22 columns]
```

[14]:
```
#Encoding categorical data: Convert categorical variables into numerical␣
 ↪representations
#(e.g., one-hot encoding or label encoding).

categorical_columns=['Education','Sex']

CVD_DF_DT = pd.get_dummies(CVD_DF_DT, columns=categorical_columns,␣
 ↪drop_first=True)
```

[15]:
```
display(CVD_DF_DT)
```

```
        HeartDiseaseorAttack  HighBP  HighChol  CholCheck   BMI  Smoker  \
0                        0.0     1.0       1.0        1.0  40.0     1.0
1                        0.0     0.0       0.0        0.0  25.0     1.0
2                        0.0     1.0       1.0        1.0  28.0     0.0
3                        0.0     1.0       0.0        1.0  27.0     0.0
4                        0.0     1.0       1.0        1.0  24.0     0.0
…                          …       …         …          …     …       …
253675                   0.0     1.0       1.0        1.0  45.0     0.0
253676                   0.0     1.0       1.0        1.0  18.0     0.0
253677                   0.0     0.0       0.0        1.0  28.0     0.0
253678                   0.0     1.0       0.0        1.0  23.0     0.0
253679                   1.0     1.0       1.0        1.0  25.0     0.0

        Stroke  Diabetes  PhysActivity  Fruits  …  PhysHlth  DiffWalk   Age  \
0          0.0       0.0           0.0     0.0  …      15.0       1.0   9.0
1          0.0       0.0           1.0     0.0  …       0.0       0.0   7.0
2          0.0       0.0           0.0     1.0  …      30.0       1.0   9.0
3          0.0       0.0           1.0     1.0  …       0.0       0.0  11.0
4          0.0       0.0           1.0     1.0  …       0.0       0.0  11.0
…            …         …             …       …  …         …         …     …
253675     0.0       0.0           0.0     1.0  …       5.0       0.0   5.0
253676     0.0       1.0           0.0     0.0  …       0.0       1.0  11.0
253677     0.0       0.0           1.0     1.0  …       0.0       0.0   2.0
253678     0.0       0.0           0.0     1.0  …       0.0       0.0   7.0
253679     0.0       1.0           1.0     1.0  …       0.0       0.0   9.0

        Income  Education_2.0  Education_3.0  Education_4.0  Education_5.0  \
0          3.0              0              0              1              0
1          1.0              0              0              0              0
2          8.0              0              0              1              0
3          6.0              0              1              0              0
4          4.0              0              0              0              1
```

```
…          …              …              …              …              …
253675     7.0            0              0              0              0
253676     4.0            1              0              0              0
253677     2.0            0              0              0              1
253678     1.0            0              0              0              1
253679     2.0            0              0              0              0

        Education_6.0  Sex_1.0
0                   0        0
1                   1        0
2                   0        0
3                   0        0
4                   0        0
…                 …        …
253675              1        1
253676              0        0
253677              0        0
253678              0        1
253679              1        0

[229781 rows x 26 columns]
```

[16]:

```
#checking outliers and plotting graphs

def find_outliers_IQR(df):

   q1=df.quantile(0.25)

   q3=df.quantile(0.75)

   IQR=q3-q1

   outliers = df[((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]

   return outliers
```

[17]:

```
outliers = find_outliers_IQR(CVD_DF_DT)

print('number of outliers: ', str(len(outliers)))

print('max outlier value: ', str(outliers.max()))

print('min outlier value: ', str(outliers.min()))

# outliers
```

```
number of outliers:  229781
max outlier value:  HeartDiseaseorAttack     1.0
```

```
HighBP                   NaN
HighChol                 NaN
CholCheck                0.0
BMI                     98.0
Smoker                   NaN
Stroke                   1.0
Diabetes                 1.0
PhysActivity             NaN
Fruits                   NaN
Veggies                  0.0
HvyAlcoholConsump        1.0
AnyHealthcare            0.0
NoDocbcCost              1.0
GenHlth                  5.0
MentHlth                30.0
PhysHlth                30.0
DiffWalk                 1.0
Age                      NaN
Income                   NaN
Education_2.0            1.0
Education_3.0            1.0
Education_4.0            NaN
Education_5.0            NaN
Education_6.0            NaN
Sex_1.0                  NaN
dtype: float64
min outlier value:  HeartDiseaseorAttack     1.0
HighBP                   NaN
HighChol                 NaN
CholCheck                0.0
BMI                     45.0
Smoker                   NaN
Stroke                   1.0
Diabetes                 0.5
PhysActivity             NaN
Fruits                   NaN
Veggies                  0.0
HvyAlcoholConsump        1.0
AnyHealthcare            0.0
NoDocbcCost              1.0
GenHlth                  5.0
MentHlth                 6.0
PhysHlth                11.0
DiffWalk                 1.0
Age                      NaN
Income                   NaN
Education_2.0            1.0
Education_3.0            1.0
```

```
Education_4.0           NaN
Education_5.0           NaN
Education_6.0           NaN
Sex_1.0                 NaN
dtype: float64
```

[18]:
```
fig = px.box(CVD_DF_DT, y='BMI')
fig.show()
```

[19]:
```
fig = px.box(CVD_DF_DT, y='MentHlth')
fig.show()
```

[20]:
```
fig = px.box(CVD_DF_DT, y='GenHlth')
fig.show()
```

[21]:
```
fig = px.box(CVD_DF_DT, y='PhysHlth')
fig.show()
```

[22]:
```
fig = px.box(CVD_DF_DT, y='Age')
fig.show()
```

[23]:
```
#Feature engineering: Create new features or modify existing ones to enhance␣
↪the model's performance.
# for age

age_bins = [0.0, 5.0, 8.0, 11.0]  # Define your scaled age bins as needed
age_labels = ['Young', 'Adult', 'Elderly']

# Create the 'AgeCategory' column based on scaled ages
CVD_DF_DT['AgeCategory'] = pd.cut(CVD_DF_DT['Age'], bins=age_bins,␣
↪labels=age_labels ,include_lowest=True)
```

[24]:
```
CVD_DF_DT.head()
```

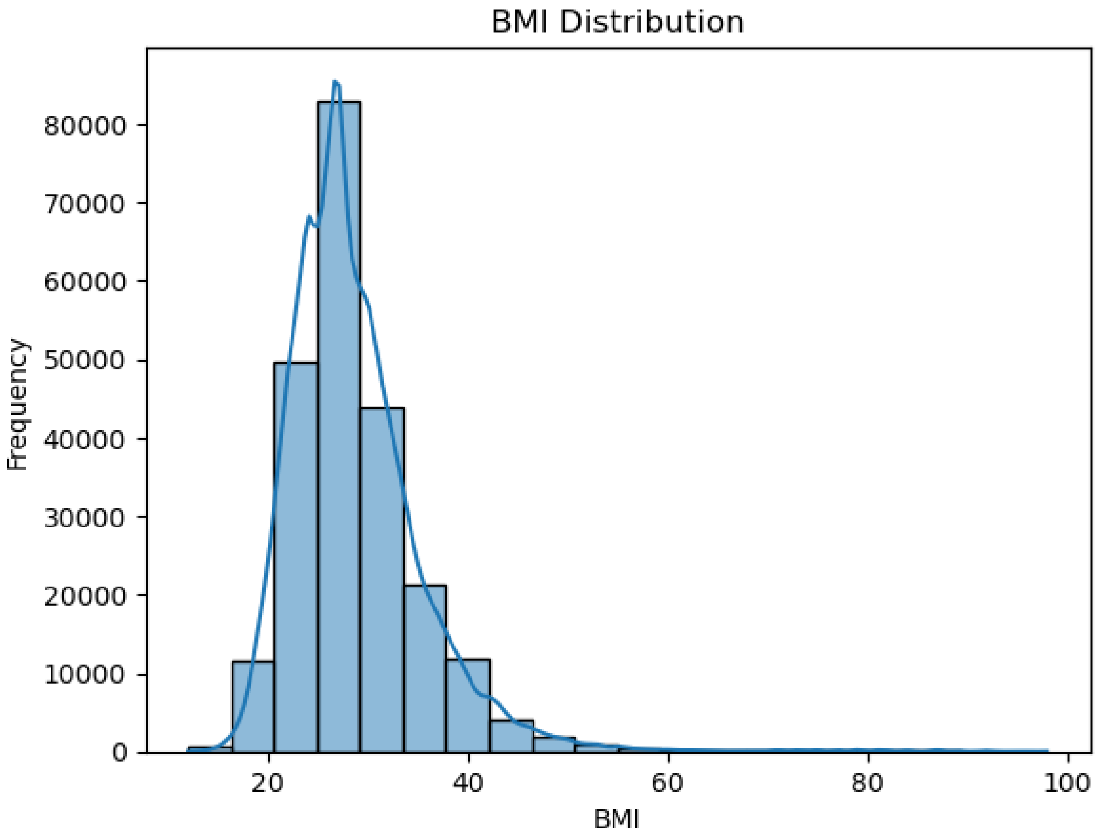
[24]:
```
   HeartDiseaseorAttack  HighBP  HighChol  CholCheck   BMI  Smoker  Stroke  \
0                   0.0     1.0       1.0        1.0  40.0     1.0     0.0
1                   0.0     0.0       0.0        0.0  25.0     1.0     0.0
2                   0.0     1.0       1.0        1.0  28.0     0.0     0.0
3                   0.0     1.0       0.0        1.0  27.0     0.0     0.0
4                   0.0     1.0       1.0        1.0  24.0     0.0     0.0

   Diabetes  PhysActivity  Fruits  …  DiffWalk   Age  Income  Education_2.0  \
0       0.0           0.0     0.0  …       1.0   9.0     3.0              0
1       0.0           1.0     0.0  …       0.0   7.0     1.0              0
2       0.0           0.0     1.0  …       1.0   9.0     8.0              0
3       0.0           1.0     1.0  …       0.0  11.0     6.0              0
4       0.0           1.0     1.0  …       0.0  11.0     4.0              0
```

```
   Education_3.0  Education_4.0  Education_5.0  Education_6.0  Sex_1.0  \
0              0              1              0              0        0
1              0              0              0              1        0
2              0              1              0              0        0
3              1              0              0              0        0
4              0              0              1              0        0

   AgeCategory
0      Elderly
1        Adult
2      Elderly
3      Elderly
4      Elderly

[5 rows x 27 columns]
```

[25]:
```python
#Feature engineering: Create new features or modify existing ones to enhance␣
↪the model's performance.

BMI_bins = [0, 18.5, 24.9, 29.9, 100]
BMI_labels = ['Underweight', 'Normal Weight', 'Overweight', 'Obese']
CVD_DF_DT['BMICategory'] = pd.cut(CVD_DF_DT['BMI'], bins=BMI_bins,␣
↪labels=BMI_labels)
```

1 Univariate Plots

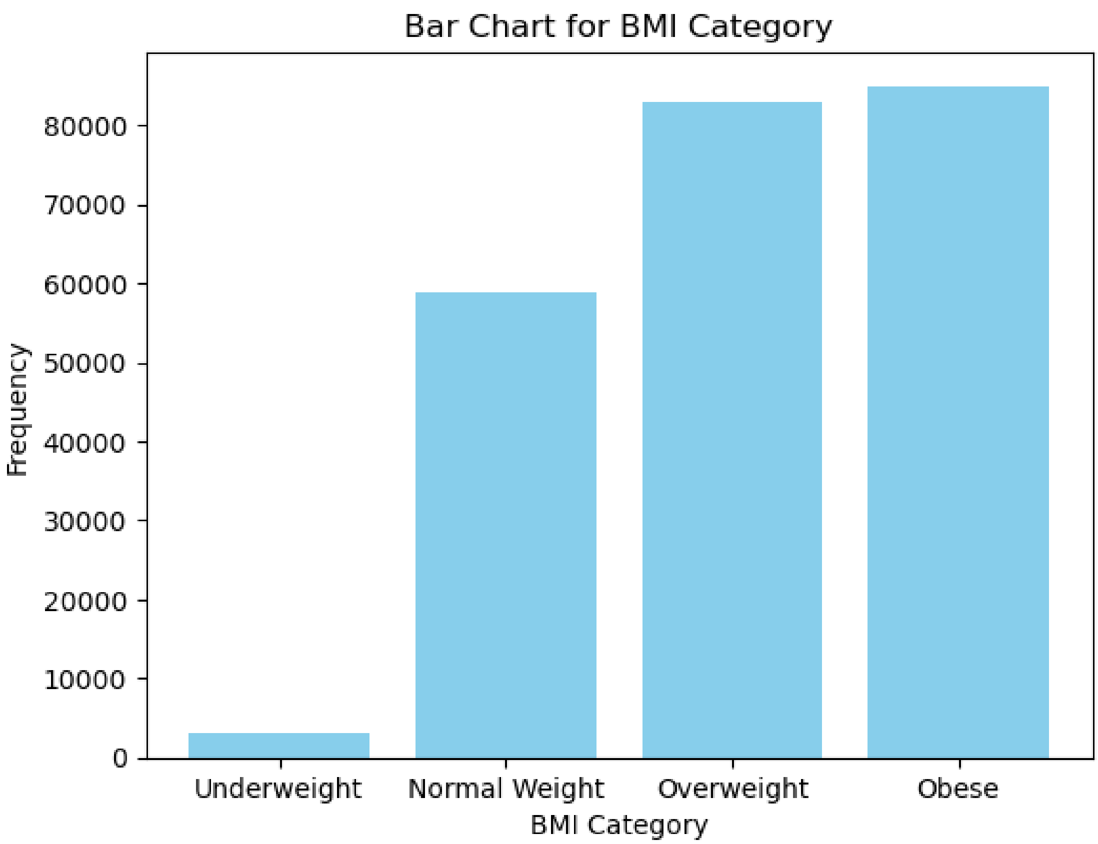
[26]:
```python
# Plot histogram for 'BMI'
sns.histplot(CVD_DF_DT, x='BMI', bins=20, kde=True)

# Update the layout
plt.xlabel("BMI")
plt.ylabel("Frequency")
plt.title("BMI Distribution")
plt.show()
```



2 Insights:

From the above Histogram of BMI Distribution, we can notice that the data is skewed towards right, that means there are more values on the left side of the distribution. The BMI is normally distributed. The median of this data is around 30. Also, we can conclude that the frequency of BMI is highest at around 30.

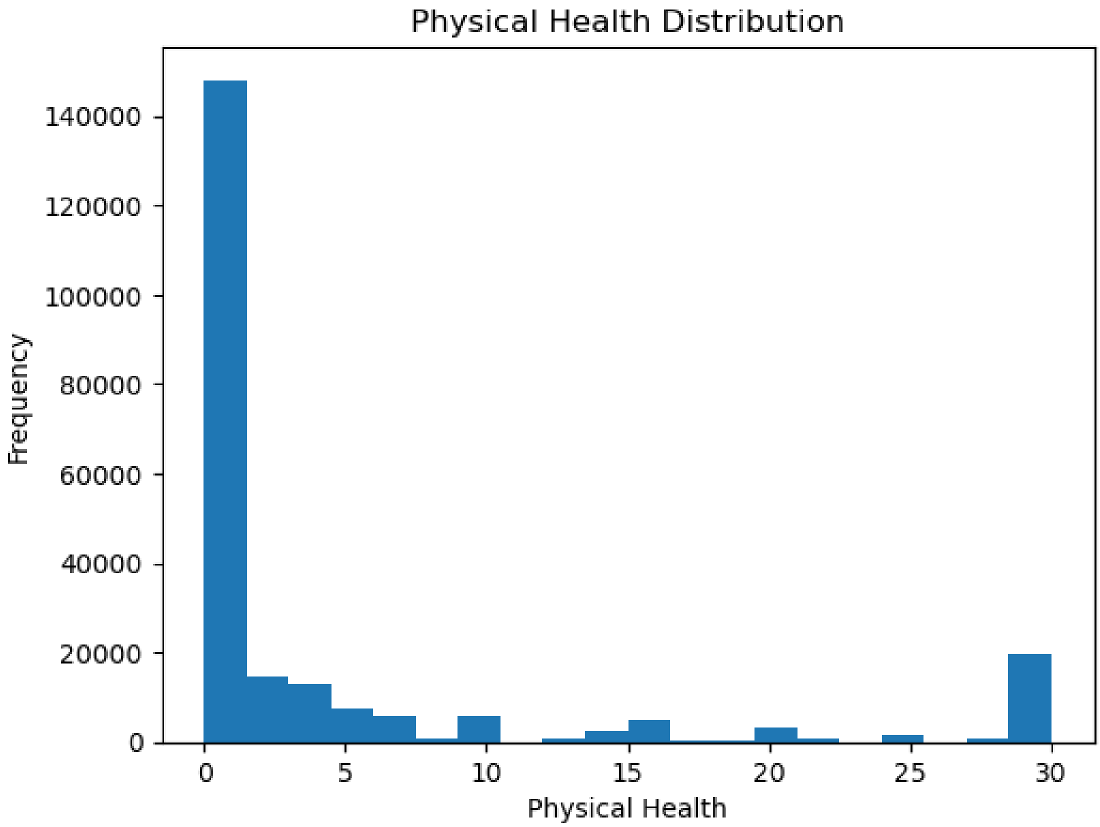
[27]:
```
# Plot bar chart for 'BMICategory'
bmi_counts = CVD_DF_DT.groupby("BMICategory").size()
plt.bar(bmi_counts.index, bmi_counts.values, color="skyblue")

# Update the layout
plt.xlabel("BMI Category")
plt.ylabel("Frequency")
plt.title("Bar Chart for BMI Category")
plt.show()
```



3 Insights:

From the above bar chart of BMI Category, we can interpret that the most common BMI Category among CVD patients is 'Obese' and 'Overweight', with about 9000 patients. The least common BMI Category is 'Underweight', with less than 500 patients. There are about 6000 patients who are of 'Normal Weight'. These insights also suggest that higher frequencies of BMI categories are associated with higher frequencies of CVD patients. However, there are other factors that would also relate to CVD.

[28]:
```
# Plot histogram for 'Physical Health'
plt.hist(CVD_DF_DT["PhysHlth"], bins=20)

# Update the layout
plt.xlabel("Physical Health")
plt.ylabel("Frequency")
plt.title("Physical Health Distribution")
plt.show()
```



4 Insights:

The above histogram depicts the Physical Health Distribution. We can notice that there are very few people, about 20000, who engage in physical activities. And greater number of people, about more than 140000 are less physically active. These insights also suggest that more the physical activity, less the risK OF CVD. But, there are other factors as well which needs to be considered.

[29]:
```
# Plot pie chart for 'AgeCategory'

counts = CVD_DF_DT["AgeCategory"].value_counts()
fig = px.pie(counts, values=counts.values, names=counts.index, title="Age␣
↪Category Distribution")
fig.show()
```

5 Insights:

From the above pie-chart, we can state that the most common age group is of “elderly”, accounting 49.2% of the total. The least common age group is of “young”, which accounts for 15.2% of the total. We may be in the delusion of elderly people more prone to CVD. But, this might not be the case always.

[30]:
```
# Plot bar chart for "Smoker"
df = CVD_DF_DT["Smoker"].value_counts().reset_index()
df.columns = ["Smoker", "Frequency"]

# Update the layout
fig = px.bar(df, x="Smoker", y="Frequency", color="Smoker",␣
 ↪color_discrete_map={"Non-smoker":"lightgreen", "Smoker":"skyblue"})
fig.update_layout(xaxis_title="Smoker/ Non-Smoker", yaxis_title="Frequency",␣
 ↪title="Bar Chart for Smoker/ Non-Smoker")
fig.show()
```

6 Insights:

The above bar chart shows the frequency of Smokers/ Non-Smokers. The number of non-smokers are more than the number of smokers. We may come to a conclusion that, smokers are at higher risk of getting CVD. But, this might not be the only case.

7 Bivariate Plots

[31]:
```
#Plot between BMI and HighBP using plotly
CVD_DF_DT_gp = CVD_DF_DT.groupby('BMICategory')['HighBP'].mean().reset_index()
fig = px.line(CVD_DF_DT_gp, x='BMICategory', y='HighBP', markers=True,␣
 ↪title='Mean HighBP by BMICategory')
fig.update_layout(xaxis_title='BMICategory', yaxis_title='Mean HighBP')
fig.show()
```

8 Insights:

This bar chart illustrates how high blood pressure levels change between weight categories in an easy-to-understand visual manner, indicating that high blood pressure tends to rise in underweight to obese people. As we know HighBP is major cause of heartattacks.

[32]:
```
#Plot between High Chol and BMI Category
CVD_DF_DT_gp = CVD_DF_DT.groupby('BMICategory')['HighChol'].mean().reset_index()
fig = px.bar(CVD_DF_DT_gp, x='BMICategory', y='HighChol', title='Mean HighChol␣
 ↪by BMICategory')
fig.update_layout(xaxis_title='BMICategory', yaxis_title='Mean HighChol')
fig.show()
```

9 Insights:

This easily understood bar chart illustrates the correlation between weight categories and elevated cholesterol levels. According to the chart, people who are underweight to obese typically have higher Cholestrol Levels. Higher level of Cholestrol is also a major cause of heart attacks.

[33]:
```
#plot between age category and High Colestrol fields
CVD_DF_DT_gp = CVD_DF_DT.groupby('AgeCategory')['HighChol'].mean().reset_index()
fig = px.sunburst(
    CVD_DF_DT_gp,
    path=['AgeCategory'],
    values='HighChol',
    title='Mean HighChol Distribution by AgeCategory',
    labels={'HighChol': 'Mean HighChol'},
    hover_data={'HighChol': ':.2f'},

    color='HighChol',
    color_continuous_scale='Viridis',
)

fig.update_layout(
    margin=dict(l=0, r=0, t=40, b=0),
    coloraxis_colorbar=dict(title='Mean HighChol'),
)

fig.show()
```

10 Insights:

The prevalence of elevated cholesterol levels among various age groups is seen in this pie chart. Based on cholesterol levels, the elderly have the biggest share, followed by adults and young people.

[34]:
```
#bar chart between smoker and stroke columns
CVD_DF_DT_gp = CVD_DF_DT.groupby('Smoker')['Stroke'].mean().reset_index()

fig = px.bar(
    CVD_DF_DT_gp,
    x='Smoker',
    y='Stroke',
    color='Smoker',
    labels={'Stroke': 'Mean Stroke'},
    title='Mean Stroke by Smoking Status',
    color_continuous_scale='Cividis'
)

fig.show()
```

11 Insights:

In summary, the decision to smoke significantly elevates the risk of stroke due to the detrimental impact of tobacco on the cardiovascular system, while non-smokers typically benefit from a lower stroke risk by avoiding these harmful substances. Therefore, quitting smoking or not starting in the first place can be a crucial step toward reducing the risk of stroke and improving overall health.

[35]:
```
#line plot for fruits and veggies based on age
CVD_DF_DT_gp = CVD_DF_DT.groupby('Age')['Fruits'].sum().reset_index()

fig = px.line(
    CVD_DF_DT_gp,
    x='Age',
    y='Fruits',
    title='Line Plot of Age against Fruits',
    labels={'Fruits': 'Total Fruits Intake'},
)

fig.show()

CVD_DF_DT_gp = CVD_DF_DT.groupby('Age')['Veggies'].sum().reset_index()
fig = px.line(
    CVD_DF_DT_gp,
    x='Age',
    y='Veggies',
    title='Line Plot of Age against Fruits',
    labels={'Veggies': 'Total Veggies Intake'},
)

fig.show()
```

12 Insights

Fruit Intake: Positive correlation: Fruit intake tends to increase with age. Low intake for young children: Individuals under 2 years of age show relatively low fruit intake. Dip in adolescence: A decrease in fruit intake is noted between ages 10-12, followed by a gradual increase from age 12 onward.

Vegetable Intake: Positive correlation: With increasing age, there is a corresponding increase in vegetable intake. Low intake for young children: Vegetable intake is notably low for individuals under 2 years of age. Dip in adolescence: A dip in vegetable intake is observed between ages 10-12, followed by a gradual increase from age 12 onward.

[36]:
```
#bar plot for fruits and veggies based on BMI

CVD_DF_DT_gp = CVD_DF_DT.groupby('BMI')['Fruits'].sum().reset_index()
# Assuming CVD_DF_DT_gp is your grouped DataFrame
fig = px.bar(
    CVD_DF_DT_gp,
    x='BMI',
    y='Fruits',
```

```
    title='Bar Plot of BMI against Total Fruits Intake',
    labels={'Fruits': 'Total Fruits Intake'},
)

# Customize layout for better readability
fig.update_layout(
    xaxis=dict(tickangle=45),  # Set the rotation angle here
    xaxis_title='BMI',
    yaxis_title='Total Fruits Intake',
)

fig.show()


CVD_DF_DT_gp = CVD_DF_DT.groupby('BMI')['Veggies'].sum().reset_index()
# Assuming CVD_DF_DT_gp is your grouped DataFrame
fig = px.bar(
    CVD_DF_DT.groupby('BMI')['Veggies'].sum().reset_index(),
    x='BMI',
    y='Veggies',
    title='Bar Plot of BMI against Total Veggies Intake',
    labels={'Veggies': 'Total Veggies Intake'},
)

# Customize layout for better readability
fig.update_layout(
    xaxis=dict(tickangle=45),  # Set the rotation angle here
    xaxis_title='BMI',
    yaxis_title='Total Veggies Intake',
)

fig.show()
```

13 Insights

Fruit Intake: Lower BMI is associated with lower fruit intake. Individuals with a healthy BMI generally consume more fruits. The majority of people maintain a healthy BMI.

Vegetable Intake: Lower BMI is linked to reduced vegetable intake. Those with a healthy BMI typically have higher vegetable consumption. Overall, the majority of the population maintains a healthy BMI. In both cases, there's a consistent association between BMI and dietary habits, emphasizing the positive relationship

14 Multivariate Plots

[37]:
```
#Scatter plot visualization of the age catagory and the body mass index of the␣
↪personel.

fig = px.scatter(
    CVD_DF_DT,
    x='AgeCategory',
    y='BMI',
    color='HeartDiseaseorAttack',
    title='Scatter Plot of Age vs BMI with Heart Disease/Attack',
)
fig.show()
```

15 Insights:

From the above scatterplot, It is found evident that the elderly people are more prone to (or) has high risk of getting a heart disease. There are people in the "Adult" catagory with the risk of suffering from a heart disease due to various other factors such as physical activity, lifestyle etc in the forth-coming visualisations.

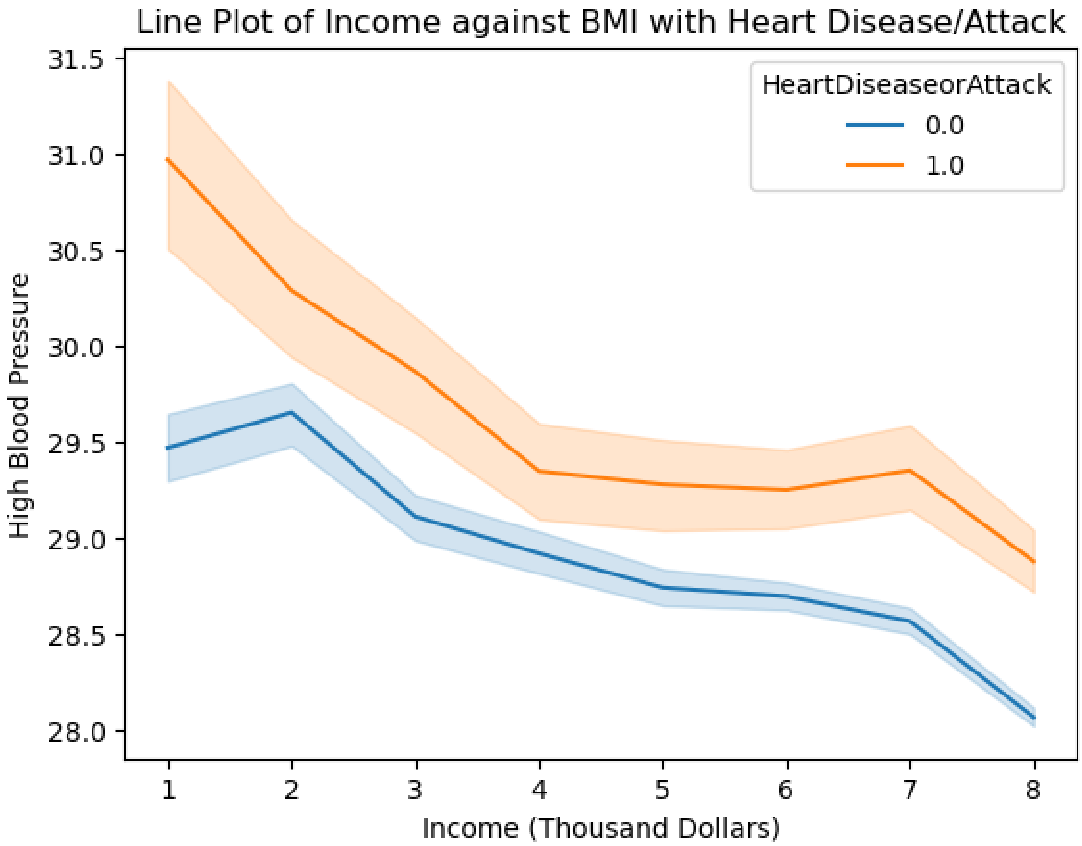
[38]:
```
#Bar chart for the individuals of both genders vulnerable to heart disease␣
↪based on BP
sns.barplot(x='Sex_1.0', y='HighBP', data=CVD_DF_DT, hue='HeartDiseaseorAttack')
plt.title('Bar Plot of Sex by HighBP with Heart Disease/Attack')
plt.xlabel('Sex (0- Female & 1- Male)')
plt.ylabel('High Blood Pressure')
plt.show()
print("Heart Disease Attack: 0 - No risk , 1- High risk")
```


```
Heart Disease Attack: 0 - No risk , 1- High risk
```

16 Insights:

Its found that the risk of getting a cardiovascular disease is irrespective of gender but is directly proportionate to the constant high blood pressure of the individual but just that women are more susceptible to the disease of the same level of blood pressure than men.

[39]:
```
# Line Plot of Income against BMI with Heart Disease/Attack
sns.lineplot(x='Income', y='BMI', data=CVD_DF_DT, hue='HeartDiseaseorAttack')
plt.title('Line Plot of Income against BMI with Heart Disease/Attack')
plt.xlabel('Income (Thousand Dollars)')
plt.ylabel('High Blood Pressure')
plt.show()
```



17 Insights:

The above graph is showing an downward trend where the individuals having high blood pressure tend to suffer from more of heart disease for the people with less income.

[40]:
```
#Interactive Scatterplot for BMI vs Physical activity
fig = px.scatter(CVD_DF_DT, x='PhysActivity', y='BMI',␣
↪color='HeartDiseaseorAttack',
                 title='Scatter Plot of PhysActivity against BMI with Heart␣
↪Disease/Attack',
                 labels={'PhysActivity': 'Physical Activity (0-No activity &␣
↪1-Highly active)', 'BMI': 'Body Mass Index'},
                 hover_data=['HeartDiseaseorAttack'])

fig.show()
```

18 Insights:

Individuals with no physical activity and are above 25 BMI (Normal catagory) are more vulnerable to getting a heart disease. It can also be inferred that the Body mass index has a strong positive correlation with the expectancy of heart disease as it can be found that people who are active yet have a higher BMI(<25) are also succeptible.

[41]:
```
# Boxplot of AgeCategory by BMI with Heart Disease/Attack
fig = px.box(
    CVD_DF_DT,
    x='AgeCategory',
    y='BMI',
    color='HeartDiseaseorAttack',
    title='Boxplot of AgeCategory by BMI with Heart Disease/Attack',
    labels={'AgeCategory': 'Age Category', 'BMI': 'BMI'},
)
fig.show()
```

19 Insights:

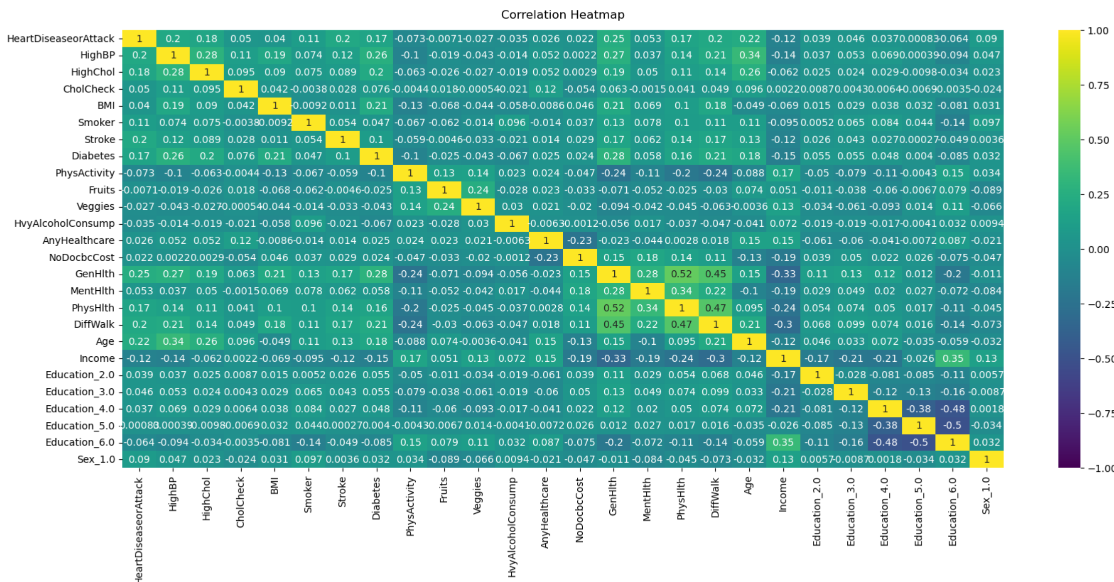
Analyzing a box plot of age groups based on BMI exposes a noteworthy trend: the majority of individuals falling within the 27–28 BMI range appear to have a relatively lower risk of experiencing a heart attack. This observation holds even as the likelihood of heart attacks tends to increase with age, emphasizing the importance of BMI as a potential mitigating factor in heart attack risk across different age groups.

[42]:
```
import plotly.graph_objects as go
# Creating an intereactive box plot using Plotly Graph Objects
fig = go.Figure()

for heart_disease_status, data in CVD_DF_DT.groupby('HeartDiseaseorAttack'):
    fig.add_trace(go.Box(x=data['AgeCategory'], y=data['BMI'],
                         name=str(heart_disease_status), boxpoints="all",␣
 ↪jitter=0.3))

fig.update_layout(
    title='Boxplot of AgeCategory by BMI with Heart Disease/Attack',
    xaxis=dict(title='Age Category'),
    yaxis=dict(title='Body Mass Index'),
)

print("")

# Show the Plotly Graph Objects plot
fig.show()
```

20 Insights:

It is found that the median value of the BMI is less for individuals across age catagories with less /no risk of heart disease wich further strengthens the fact that it has a direct correlation.

[40]:
```
# Explore corrleations between features
plt.figure(figsize=(20,8))
heatmap = sns.heatmap(CVD_DF_DT.corr(),vmin=-1, vmax=1, annot=True,
    cmap='viridis');
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':12}, pad=12);
```



21 Insights:

Correlations with respect to possibility of getting a heart disease 1) A heavy positive correlation of BMI and has found to be a major factor impacting the possible outcome. 2) Age and physical health also is directly proportional affecting the scenario.

[ ]:
```
```