CSE 587 **Fall 2022**

Project Phase #1

Abhi Vijayakumar and **Harshitha Siddaramaiah**

Problem Statement: Prediction of Heart Disease using patient examination results.

This project focuses on analysing the patient examination results and whether engaging in regular physical activity, maintaining a healthy diet and weight, managing stress, avoiding smoking, and getting quality sleep each night can all reduce the risk of heart disease and help people live longer, healthier lives.

Driving Force:

According to the Centres for Disease Control and Prevention (CDC), heart disease is one of the leading causes of death in the United States, claiming the lives of more than 650,000 people each year. Even though the research and innovation over the recent years have made considerable progress to treat heart diseases, it continues to exact a heart-breaking toll. People suffering from heart disease and related conditions are also at increased risk of severe illness and long-term effects from COVID-19. Cardiovascular diseases are also a leading cause of pregnancy-related deaths, which are highest among women of colour. These heart health disparities have been related to several socioeconomic, environmental, and social variables. Our attempt is to analyse, identify and address such tragic disparities to improve heart health.

Project Goal:

About half the population of Americans have at least 1 of the 3 key risk factors for heart disease: high blood pressure, high cholesterol, and smoking. Other key indicators include diabetic status, obesity (high BMI), not getting enough physical activity or drinking too much alcohol. Our project aims at detecting those factors that have the greatest impact on heart disease. The outcome of this project could serve as important information in healthcare to raise awareness and the actions we can all take to prevent heart diseases. Our project can be

considered as a starting point that could potentially help bigger research projects capable of investing billions of dollars in preventing, detecting, treating cardiovascular conditions, and to develop new programs to alleviate heart health disparities. Overall, in this project we are doing our bit to ensure a healthier future for humanity.

Data Source:

We collected the above (attached) data set from the Kaggle site. Originally, this data set is from CDC and is a major part of the Behavioural Risk Factor Surveillance System (BRFSS), which conducts annual telephone surveys to gather data on the health status of the U.S. residents. BRFSS completes more than 400,000 adult interviews each year. The most recent dataset (as of February 15, 2022) consists of 401,958 rows and 279 columns. The columns in the data source are questions asked to patients about their health status, such as "How many hours do you sleep per day?" or "Have you smoked or drunk alcohol in your entire life?" In this dataset, we noticed many different factors that directly or indirectly influence heart disease, hence we decided to select some of the most relevant variables from it and did some pre-processing on it which will be discussed in the further sections in the report. The below screenshots depict the data set we have.

```
In [41]:
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sea
```

```
In [42]:
Location = r'C:\Users\abhi\Documents\heart_2020_Cleaned.csv'
df2 = pd.read_csv(Location)
```

```
In [43]:
df2
```

Out[43]:

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	Physic
0	No	16.60	Yes	No	No	3.0	30	No	Female	55-59	White	Yes	
1	No	20.34	No	No	Yes	0.0	0	No	Female	80 or older	White	No	
2	No	26.58	Yes	No	No	20.0	30	No	Male	65-69	White	Yes	
3	No	24.21	No	No	No	0.0	0	No	Female	75-79	White	No	
4	No	23.71	No	No	No	28.0	0	Yes	Female	40-44	White	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	
319800	No	26.58	Yes	No	No	0.0	0	Yes	Male	NaN	Hispanic	No	
319801	No	31.93	No	Yes	No	0.0	0	NaN	Male	NaN	Hispanic	No	
319802	Yes	33.20	Yes	No	No	0.0	0	No	Female	60-64	Hispanic	Yes	
319803	No	36.54	No	No	No	7.0	0	No	Male	30-34	Hispanic	No	
319804	No	23.38	No	No	No	0.0	0	No	Female	60-64	Hispanic	No	

319805 rows × 18 columns

In [44]:
```
df2.head() #Data Overview
```

Out[44]:

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity
0	No	16.60	Yes	No	No	3.0	30	No	Female	55-59	White	Yes	Ye
1	No	20.34	No	No	Yes	0.0	0	No	Female	80 or older	White	No	Ye
2	No	26.58	Yes	No	No	20.0	30	No	Male	65-69	White	Yes	Ye
3	No	24.21	No	No	No	0.0	0	No	Female	75-79	White	No	N
4	No	23.71	No	No	No	28.0	0	Yes	Female	40-44	White	No	Ye

In [45]:
```
df2.dtypes
```

Out[45]:
```
HeartDisease         object
BMI                 float64
Smoking              object
AlcoholDrinking      object
Stroke               object
PhysicalHealth      float64
MentalHealth          int64
DiffWalking          object
Sex                  object
AgeCategory          object
Race                 object
Diabetic             object
PhysicalActivity     object
GenHealth            object
SleepTime             int64
Asthma               object
KidneyDisease        object
SkinCancer           object
dtype: object
```

Pre-processing Techniques:

1. Checking for Null values:

The data set for heart analysis consists of around 319,805 rows and 18 columns. The null values in each column are checked and those rows that have null values are dropped as the first step.

In [46]:
```
#Check for the null values in each column data
print(df2.isna().sum()) #returns the number of missing values in each column
```

```
HeartDisease        1
BMI                 1
Smoking             1
AlcoholDrinking     1
Stroke              0
PhysicalHealth      1
MentalHealth        0
DiffWalking         2
Sex                 0
AgeCategory         2
Race                0
Diabetic            0
PhysicalActivity    0
GenHealth           2
SleepTime           0
Asthma              0
KidneyDisease       0
SkinCancer          0
dtype: int64
```

In [47]:
```
df2 = df2.dropna() #dropping all null valued columns
df2
```

Out[47]:

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	Physic
0	No	16.60	Yes	No	No	3.0	30	No	Female	55-59	White	Yes	
1	No	20.34	No	No	Yes	0.0	0	No	Female	80 or older	White	No	
2	No	26.58	Yes	No	No	20.0	30	No	Male	65-69	White	Yes	
3	No	24.21	No	No	No	0.0	0	No	Female	75-79	White	No	
4	No	23.71	No	No	No	28.0	0	Yes	Female	40-44	White	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
319790	Yes	27.41	Yes	No	No	7.0	0	Yes	Male	60-64	Hispanic	Yes	
319791	No	29.84	Yes	No	No	0.0	0	No	Male	35-39	Hispanic	No	
319792	No	24.24	No	No	No	0.0	0	No	Female	45-49	Hispanic	No	
319793	No	32.81	No	No	No	0.0	0	No	Female	25-29	Hispanic	No	
319803	No	36.54	No	No	No	7.0	0	No	Male	30-34	Hispanic	No	

319795 rows × 18 columns

2. Checking for Duplicate values:

In the next step, we are checking if there are any duplicate rows in the table, if there are, then those rows are deleted.

```
In [78]:  #Check for duplicated values
          df2.duplicated(keep=False)
```

```
Out[78]: 0         False
         1         False
         2         False
         3         False
         4         False
                   ...
         319790    False
         319791    False
         319792    False
         319793    False
         319794    False
         Length: 319795, dtype: bool
```

3. Binary encoding:

All the object type variables in the columns are encoded to 0 or 1. If the column values contain Yes, then they are encoded as 1 and 0 for No. For the Sex column, Male is encoded as 1 and Female as 0. The following columns in the data set were encoded to Binary values:

- HeartDisease
- Sex
- Smoking
- AlcoholDrinking
- Stroke
- DiffWalking
- PhysicalActivity
- Asthma
- KidneyDisease
- SkinCancer

```
In [121]: #Binary encoding
          df2['HeartDisease'] = df2['HeartDisease'].replace({'Yes':1, 'No':0})
          df2['Sex'] = df2['Sex'].replace({'Female':0, 'Male':1})
          df2['Smoking'] = df2['Smoking'].replace({'Yes':1, 'No':0})
          df2['AlcoholDrinking'] = df2['AlcoholDrinking'].replace({'Yes':1, 'No':0})
          df2['Stroke'] = df2['Stroke'].replace({'Yes':1, 'No':0})
          df2['DiffWalking'] = df2['DiffWalking'].replace({'Yes':1, 'No':0})
          df2['PhysicalActivity'] = df2['PhysicalActivity'].replace({'Yes':1, 'No':0})
          df2['Asthma'] = df2['Asthma'].replace({'Yes':1, 'No':0})
          df2['KidneyDisease'] = df2['KidneyDisease'].replace({'Yes':1, 'No':0})
          df2['SkinCancer'] = df2['SkinCancer'].replace({'Yes':1, 'No':0})
          #df2['Sex'].value_counts
          df2.dtypes
```

```
Out[121]: HeartDisease          int64
          BMI                 float64
          Smoking               int64
          AlcoholDrinking       int64
          Stroke                int64
          PhysicalHealth      float64
          MentalHealth        float64
          DiffWalking           int64
          Sex                   int64
          AgeCategory          object
          Race                 object
          Diabetic             object
          PhysicalActivity      int64
          GenHealth            object
          SleepTime           float64
          Asthma                int64
          KidneyDisease         int64
          SkinCancer            int64
          dtype: object
```

4. Converting object datatype to string:

Before categorical encoding, the object data type must be converted to string data type so that values could be encoded to different integer values. Hence, converting the below columns in the data set from "object" data type to "string" data type:

- AgeCategory
- Race
- Diabetic
- GenHealth

In [122]:
```
#Change object dtype to string
df2['AgeCategory'] = df2['AgeCategory'].astype("string")
df2['Race'] = df2['Race'].astype("string")
df2['Diabetic'] = df2['Diabetic'].astype("string")
# df2['PhysicalActivity'] = df2['PhysicalActivity'].astype("string")
df2['GenHealth'] = df2['GenHealth'].astype("string")
df2.dtypes
```

Out[122]:
```
HeartDisease          int64
BMI                 float64
Smoking               int64
AlcoholDrinking       int64
Stroke                int64
PhysicalHealth      float64
MentalHealth        float64
DiffWalking           int64
Sex                   int64
AgeCategory          string
Race                 string
Diabetic             string
PhysicalActivity      int64
GenHealth            string
SleepTime           float64
Asthma                int64
KidneyDisease         int64
SkinCancer            int64
dtype: object
```

5. Categorical encoding:

After converting the above columns to string data type, the values in the columns are encoded to different categorical integer values and added as a new column. If there are any other outlier values other than the values within the expected range, are set to a higher value 100 (refer the attached code below). With the categorical encoding, the correlation matrix can be defined which further removes the outliers. The following columns in the data set were encoded to Categorical values:

- AgeCategory
- Diabetic
- Race
- GenHealth

```
def AgeCheck(x):
    if x =='18-24':
        return 0
    elif x=='25-29':
        return 1
    elif x=='30-34':
        return 2
    elif x=='35-39':
        return 3
    elif x=='40-44':
        return 4
    elif x=='45-49':
        return 5
    elif x=='50-54':
        return 6
    elif x=='55-59':
        return 7
    elif x=='60-64':
        return 8
    elif x=='65-69':
        return 9
    elif x=='70-74':
        return 10
    elif x=='75-79':
        return 11
    elif x=='80 or older':
        return 12
    else:
        return 100
df2['AgeCategory_Category'] = df2['AgeCategory'].apply(AgeCheck)


def DiabeticCheck(x):
    if x=="No":
        return 0
    elif x=="Yes":
        return 1
    elif x=="No, borderline diabetes":
        return 2
    elif x=="Yes (during pregnancy)":
        return 3
    else:
        return 100
df2['Diabetic_Category'] = df2['Diabetic'].apply(DiabeticCheck)

def RaceCheck(x):
    if x=="White":
        return 1
    elif x=="Black":
        return 2
    elif x=="Hispanic":
        return 3
    elif x=="Asian":
        return 4
    elif x=="American Indian/Alaskan Native":
```

```
    elif x=="American Indian/Alaskan Native":
        return 5
    elif x=="Other":
        return 6
    else:
        return 100
df2['Race_Category'] = df2['Race'].apply(RaceCheck)

def GenCheck(x):
    if x=="Very good":
        return 1
    elif x=="Good":
        return 2
    elif x=="Excellent":
        return 3
    elif x=="Fair":
        return 4
    elif x=="Poor":
        return 5
    else:
        return 100
df2['GenHealth_Category'] = df2['GenHealth'].apply(GenCheck)

df2
```

10]:

PhysicalActivity	GenHealth	SleepTime	Asthma	KidneyDisease	SkinCancer	AgeCategory_Category	Diabetic_Category	Race_Category	GenHealth_Category
1	Very good	5.0	1	0	1	7	1	1	1
1	Very good	7.0	0	0	0	12	0	1	1
1	Fair	8.0	1	0	0	9	1	1	4
0	Good	6.0	0	0	1	11	0	1	2
1	Very good	8.0	0	0	0	4	0	1	1
...	...	...	...	...	...	...	...	...	...
0	Fair	6.0	1	0	0	8	1	3	4
1	Very good	5.0	1	0	0	3	0	3	1
1	Good	6.0	0	0	0	5	0	3	2
0	Good	12.0	0	0	0	1	0	3	2
1	Good	8.0	0	0	0	12	0	3	2

6. Removing Outliers:

For the newly created columns from categorical encoding, the outlier values that were set to 100 are removed as below. The outliers were removed for the following new columns:

- AgeCategory_Category
- Diabetic_Category
- Race_Category

In [13]:
```
#Removing Outliers
indexAge = df2[ (df2['AgeCategory_Category']==100 )].index
#print(indexAge)
print(df2.drop(indexAge , inplace=True))
#print(indexAge)
indexDiabetic = df2[ (df2['Diabetic_Category'] ==100)].index
print(df2.drop(indexDiabetic , inplace=True))
indexRace = df2[ (df2['Race_Category'] ==100)].index
print(df2.drop(indexRace , inplace=True))
#df.head(15)
```

```
None
None
None
```

7. Adding a new column "BMI_Normalcy":

Based on the health data, if the BMI range is less than 18.5 or greater than 30.5, the BMI is considered Abnormal. A new column called **BMI_Normalcy** is created and based on the BMI value, the value is tagged as Normal or Abnormal.

In [126]:
```
# Add a new column named 'BMI_Normalcy'
df2['BMI_Normalcy'] = ["Abnormal" if ((x < 18.5) | (x>30.5)) else "Normal" for x in df2['BMI']]
df2 = df2.astype({'BMI_Normalcy':'string'})
#df2.BMI_Normalcy.dtype
BMI_count = df2['BMI_Normalcy'].value_counts()
#print(df2[['BMI','BMI_Normalcy']])
#df2.dtypes
print(BMI_count)
df2
```

```
Normal      220252
Abnormal     99543
Name: BMI_Normalcy, dtype: Int64
```

8. Checking if string values have any whitespaces:

For the string values, the leading and trailing whitespaces from the data are checked and if there are any whitespaces, then they are removed.

In [124]:
```
#Remove the whitespaces
df2['GenHealth'] = df2['GenHealth'].str.strip()  # or .replace as above
df2['Diabetic'] = df2['Diabetic'].str.strip()
df2['AgeCategory'] = df2['AgeCategory'].str.strip()
print(df2['Race'].str.strip())
```

```
0            White
1            White
2            White
3            White
4            White
           ...
319790    Hispanic
319791    Hispanic
319792    Hispanic
319793    Hispanic
319794    Hispanic
Name: Race, Length: 319795, dtype: string
```

9. Rounding off float values to 2 decimal places:

The columns that are of float data type are rounded to 2 decimal places.

```
In [51]: df2['BMI'].round(2)

Out[51]: 0         16.60
         1         20.34
         2         26.58
         3         24.21
         4         23.71
                   ...
         319790    27.41
         319791    29.84
         319792    24.24
         319793    32.81
         319794    46.56
         Name: BMI, Length: 319795, dtype: float64
```

10. Removing irrelevant features (after plotting Correlation matrix):

From the correlation matrix, it was evident that Physical activity, Alcohol consumption, Race and Sleep Time do not contribute to heart disease. Hence, these columns are dropped prior to the modelling.

```
In [262]: df2.drop(['AlcoholDrinking','PhysicalActivity','GenHealth', 'SleepTime'],axis=1,inplace=True)
          df2.head(5)
```

Out[262]:

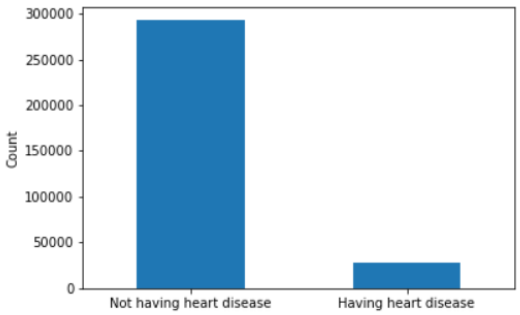
	HeartDisease	BMI	Smoking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	Asthma	KidneyDisease	SkinCanc
0	0	16.60	1	0	3.0	30.0	0	0	55-59	White	Yes	1	0	
1	0	20.34	0	1	0.0	0.0	0	0	80 or older	White	No	0	0	
2	0	26.58	1	0	20.0	30.0	0	1	65-69	White	Yes	1	0	
3	0	24.21	0	0	0.0	0.0	0	0	75-79	White	No	0	0	
4	0	23.71	0	0	28.0	0.0	1	0	40-44	White	No	0	0	

Exploratory Data Analysis:

1. How many people are suffering from heart disease?

With the pre-processed data in hand, the number of people who have a heart disease and who don't have a heart disease is plotted in the below graph. It is seen that the number of people who don't have a heart disease is higher than the number of people who have a heart disease.

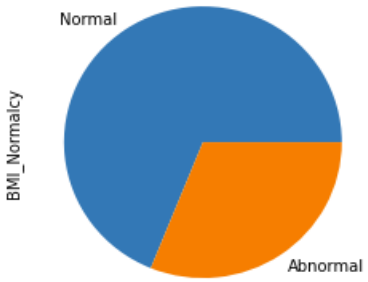
```
In [32]: heartdisease_count = df2['HeartDisease'].value_counts()
         # #heartdisease_count.plot.(kind="bar")
         # df2.plot.bar(x='heartdisease_count', rot=0);
         fig = heartdisease_count.plot(kind = 'bar')
         fig.set_xticklabels(labels=["Not having heart disease", 'Having heart disease'], rotation=0.1);
         #plt.title("Heart Disease values Count")
         plt.ylabel("Count");
```



2. How many people have Normal BMI and Abnormal BMI?

A new column BMI_Normalcy was added to the dataset during pre-processing. According to the health data, BMI range less than 18.5 and greater than 25.5 is considered as Abnormal. BMI range between 18.5 and 25.5 is considered Normal. This data of Normal vs Abnormal BMI is plotted below as a pie chart.

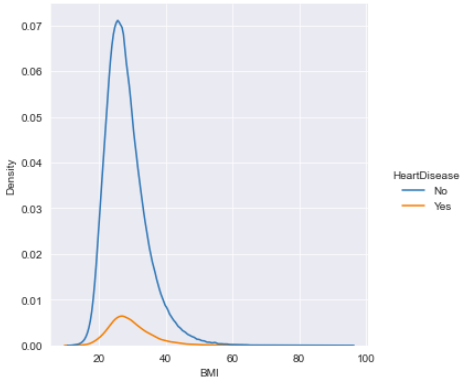
```
Name: BMI, Length: 3604, dtype: int64

In [28]: BMI_count.plot(kind='pie')

Out[28]: <AxesSubplot:ylabel='BMI_Normalcy'>
```



3. Whether Abnormal BMI contributes to causing a heart disease?

According to the CDC, a person having an Abnormal BMI has a high risk for heart disease. From the data, we have calculated the BMI_Normalcy using the BMI column and it is plotted against heart disease. From the graph, it is seen that people who have abnormal BMI (i.e the BMI range > 25.5 and < 18.5) have a heart disease. The Heart Disease column is normalized before plotting.

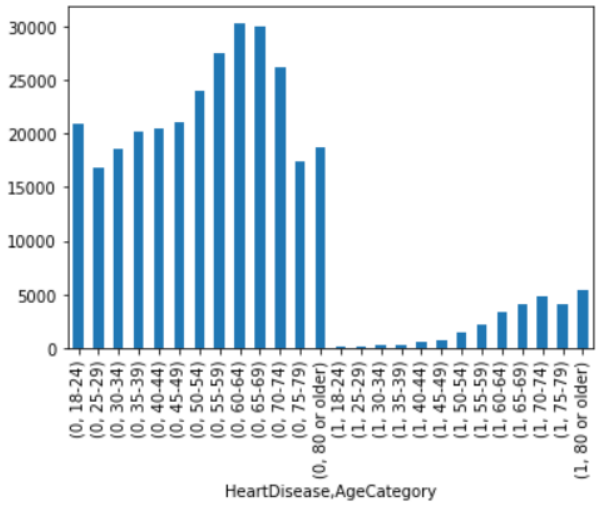
```
In [102]: #show_relation(df2['BMI'], 'HeartDisease');
          sns.displot(data=df2, x=df2['BMI'], hue=df2['HeartDisease'], kind='kde');
```



4. To find which Age Category people are more prone to get a heart disease:

While plotting the heart disease with the Age category, it is seen that people who are above the age of 60 are more prone to get a heart disease. Specifically, people who belong to the age group 70-74 have a higher risk for getting a heart disease.

```
In [44]: df2.groupby(['HeartDisease','AgeCategory']).size().plot.bar()
Out[44]: <AxesSubplot:xlabel='HeartDisease,AgeCategory'>
```



5. Comparing the ratio of Males v/s Females who consume Alcohol and checking if that has an effect on heart disease:

In the below code, Sex column indicates 0 for Females and 1 for Males. It is seen that the number of females who consume alcohol and have a heart disease is 428 and the number of males who consume alcohol and have a heart disease is 713, which is very less in number when compared to the total data set that we have. So, with the data analysis, we can conclude that Alcohol consumption doesn't contribute much towards getting a heart disease.

```
In [65]: #print(df2.groupby('HeartDisease','Sex')['AlcoholDrinking'].sum())
groups = df2.groupby('Sex')['AlcoholDrinking'].sum()
df = pd.DataFrame(groups)
print(df)
#df2.groupby('HeartDisease')(groups).sum()
print(df2.groupby(['HeartDisease','Sex','AlcoholDrinking']).size())
#print(df2.groupby('HeartDisease','Sex')['AlcoholDrinking'].sum())
```

```
     AlcoholDrinking
Sex
0              11258
1              10519
HeartDisease  Sex  AlcoholDrinking
0             0    0                145741
                   1                 10830
              1    0                126045
                   1                  9806
1             0    0                 10806
                   1                   428
              1    0                 15426
                   1                   713
dtype: int64
```

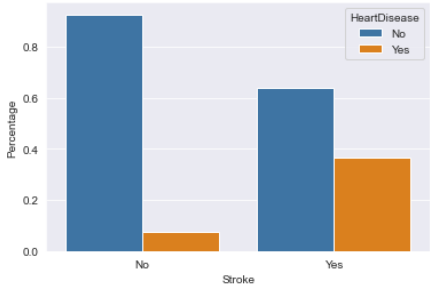
6. Checking whether other diseases contribute towards getting a heart disease:

The effect of having other diseases like Stroke increases the chances of heart disease is analysed below. The relative frequency value is considered for the analysis.

1. Stroke and Heart Disease:

In [158]:
```
perc = df2.groupby('Stroke')['HeartDisease'].value_counts(normalize=True).reset_index(name='Percentage')
print(perc)
sns.barplot(data=perc, x='Stroke',y='Percentage', hue='HeartDisease', order=df2['Stroke'].value_counts().index);
```

```
  Stroke HeartDisease  Percentage
0     No           No    0.925310
1     No          Yes    0.074690
2    Yes           No    0.636341
3    Yes          Yes    0.363659
```



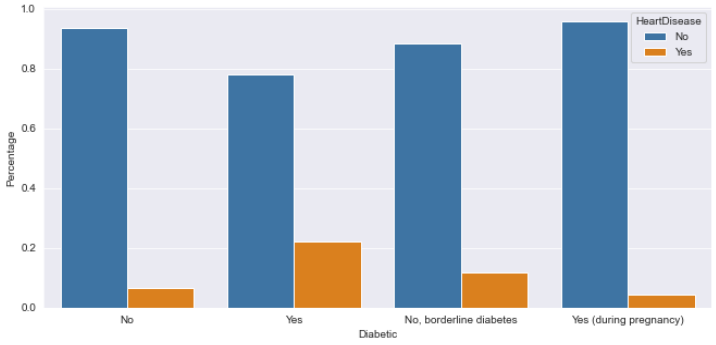
From the above graph, we can infer that Stroke is highly associated with heart disease.

2. Diabetic and Heart Disease:

From the below graph it can be inferred that Diabetes does have an effect on heart disease.

In [161]:
```
#show_relation('Diabetic', 'HeartDisease', type_='count')
plt.figure(figsize=(11,5));
perc = df2.groupby('Diabetic')['HeartDisease'].value_counts(normalize=True).reset_index(name='Percentage')
print(perc)
sns.barplot(data=perc, x='Diabetic',y='Percentage', hue='HeartDisease', order=df2['Diabetic'].value_counts().index);
```

```
                 Diabetic HeartDisease  Percentage
0                      No           No    0.935031
1                      No          Yes    0.064969
2  No, borderline diabetes           No    0.883645
3  No, borderline diabetes          Yes    0.116355
4                     Yes           No    0.780476
5                     Yes          Yes    0.219524
6   Yes (during pregnancy)           No    0.957796
7   Yes (during pregnancy)          Yes    0.042204
```

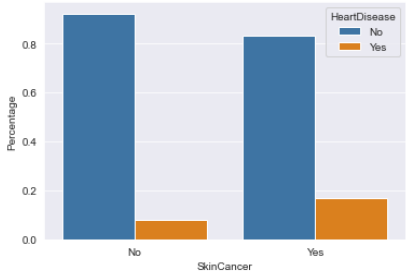


3. Skin cancer and heart disease:

From the below graph, we can say that very few people who suffer from skin cancer have a heart disease.

In [165]:
```
perc = df2.groupby('SkinCancer')['HeartDisease'].value_counts(normalize=True).reset_index(name='Percentage')
print(perc)
sns.barplot(data=perc, x='SkinCancer',y='Percentage', hue='HeartDisease', order=df2['SkinCancer'].value_counts().index);
```

```
  SkinCancer HeartDisease  Percentage
0         No           No    0.922776
1         No          Yes    0.077224
2        Yes           No    0.832992
3        Yes          Yes    0.167008
```

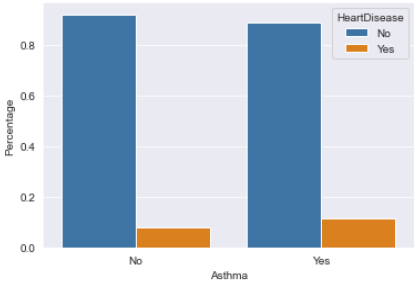


4. Asthma and heart disease:

From the below graph, we can conclude that many people who suffer from Asthma don't have a heart disease.

In [166]:
```
perc = df2.groupby('Asthma')['HeartDisease'].value_counts(normalize=True).reset_index(name='Percentage')
print(perc)
sns.barplot(data=perc, x='Asthma',y='Percentage', hue='HeartDisease', order=df2['Asthma'].value_counts().index);
```

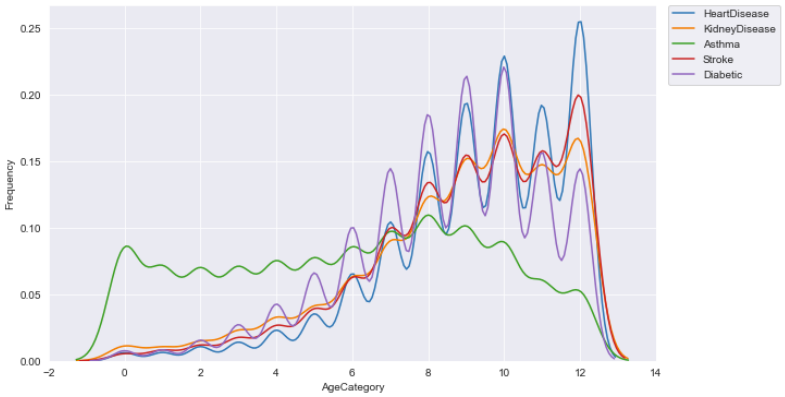
```
  Asthma HeartDisease  Percentage
0     No           No    0.918967
1     No          Yes    0.081033
2    Yes           No    0.884937
3    Yes          Yes    0.115063
```



7. Checking what age category people, who are suffering from other diseases are more prone to get a heart disease:

From the graph below, people who belong to the age category above 60 (categorical encoding having values between 8 and 12 in the x-axis) and who are diabetic are more prone to get heart diseases.

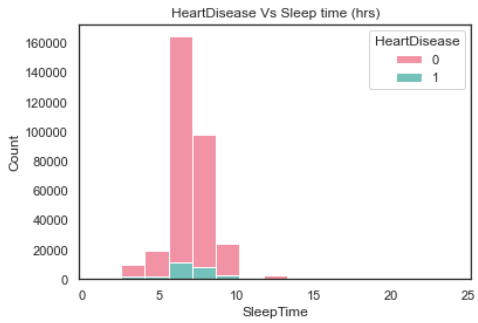
In [217]:
```
fig, ax = plt.subplots(figsize = (10,6))
sns.kdeplot(df2[df2["HeartDisease"]==1]["AgeCategory_Category"],   label="HeartDisease", ax = ax)
sns.kdeplot(df2[df2["KidneyDisease"]==1]["AgeCategory_Category"],  label="KidneyDisease", ax = ax)
#sns.kdeplot(df2[df2["SkinCancer"]==1]["AgeCategory_Category"], alpha=1,shade = False, color=colors6[2], label="SkinCancer",
sns.kdeplot(df2[df2["Asthma"]==1]["AgeCategory_Category"],   label="Asthma", ax = ax)
sns.kdeplot(df2[df2["Stroke"]==1]["AgeCategory_Category"],  label="Stroke", ax = ax)
sns.kdeplot(df2[df2["Diabetic_Category"]==1]["AgeCategory_Category"],  label="Diabetic", ax = ax)

ax.set_xlabel("AgeCategory")
ax.set_ylabel("Frequency")
ax.legend(bbox_to_anchor=(1.02, 1), loc=2, borderaxespad=0.)
plt.show()
```



8. To check if Sleep Duration has an impact on heart disease?

The sleep cycle is plotted against the heart disease column, and it is seen that people who have sleep hours ranging between 5.5 to 7 hrs have a heart disease. But the number of people who have a heart disease in that range is comparatively less.

In [237]:
```
#sns.set()
#plt.figure(figsize=(8,6))
#sns.set_style('white')
sns.histplot(x=df2['SleepTime'],hue = df2['HeartDisease'],bins=15,palette= 'husl',stat='count',multiple='stack')
#plt.xticks(list_xtickls)
plt.title('HeartDisease Vs Sleep time (hrs)')
#sns.despine()
plt.show()
```



9. Checking if Smoking influences heart disease:

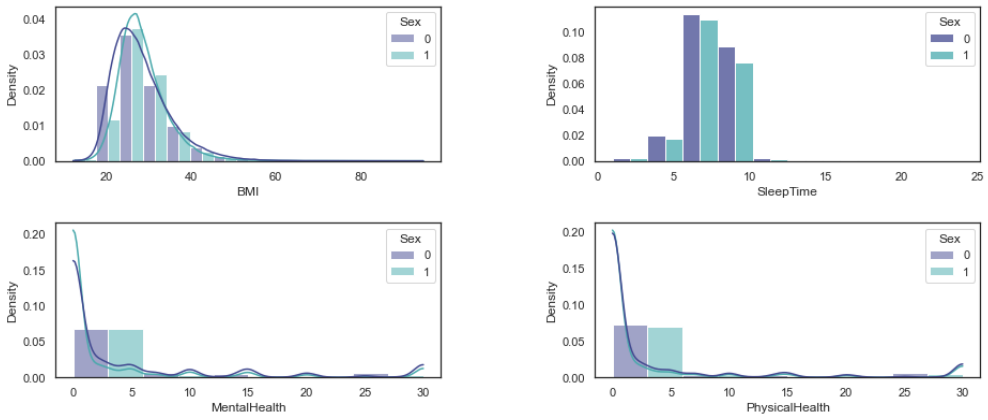
From the below data, it is seen that people who smoke are more prone to heart disease.

```
In [261]: df2.groupby('HeartDisease')['Smoking'].sum()
          # df2.groupby('Sex')['AlcoholDrinking'].sum()
          df2.groupby('Smoking')['HeartDisease'].value_counts(normalize=True)*100

Out[261]: Smoking  HeartDisease
          0        0              93.966586
                   1               6.033414
          1        0              87.842284
                   1              12.157716
          Name: HeartDisease, dtype: float64
```

10. Comparing Male and Female with factors like BMI, Mental Health, Physical health, and Sleep Time to analyse who has a healthy lifestyle:

From the below graphs, it is seen that the mean BMI for Males (Encoded with 1) is slightly higher than the Females (Encoded with 0). Both the sexes have almost the same average value for sleeping time. Distribution of Mental Health and Physical Health are almost the same too for both the genders.

```
In [251]: fig , ax = plt.subplots(2,2,figsize=(15,6))
          sns.set_style('white')
          axis = sns.histplot(x=df2[num_cols[0]],hue=df2['Sex'],stat='density',multiple='dodge',kde=True,bins=15,palette='mako',ax = ax
          sns.histplot(x=df2[num_cols[-1]],hue=df2['Sex'],stat='density',multiple='dodge',bins=10,palette='mako',ax=ax[0,1])
          sns.histplot(x=df2[num_cols[-2]],hue=df2['Sex'],stat='density',multiple='dodge',kde=True,bins=5,palette='mako',ax=ax[1,0])
          sns.histplot(x=df2[num_cols[-3]],hue=df2['Sex'],stat='density',multiple='dodge',kde=True,bins=5,palette='mako',ax=ax[1,1])
          fig.subplots_adjust(left=0.1,bottom=0.1,right=0.9,top=0.9,wspace=0.4,hspace=0.4)
```



11. Descriptive Statistics:

Below table shows the descriptive statistics like count, mean, standard deviation, min, quartiles (25%, 50% & 75%) and max for all the columns in the pre-processed data set.

```
In [185]: trans_df = df2.copy()
          trans_df.describe().T
```

Out[185]:

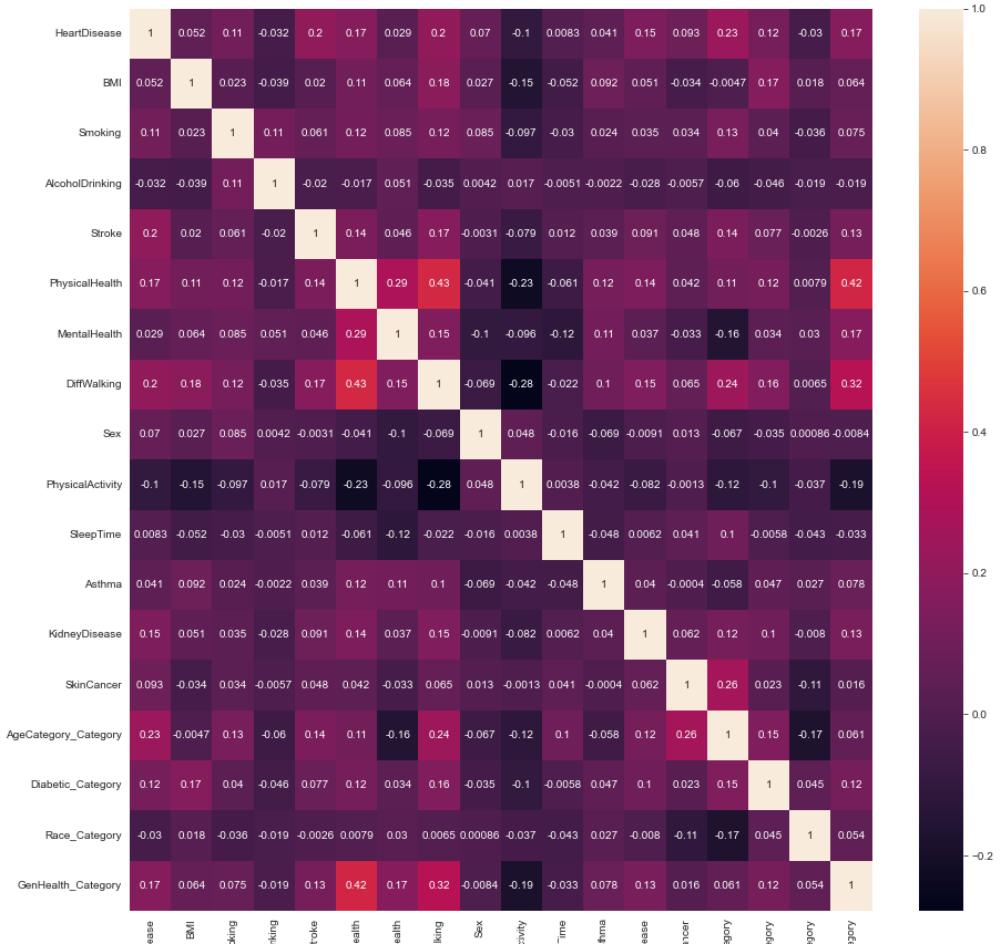
	count	mean	std	min	25%	50%	75%	max
HeartDisease	319795.0	0.085595	0.279766	0.00	0.00	0.00	0.00	1.00
BMI	319795.0	28.325399	6.356100	12.02	24.03	27.34	31.42	94.85
Smoking	319795.0	0.412477	0.492281	0.00	0.00	0.00	1.00	1.00
AlcoholDrinking	319795.0	0.068097	0.251912	0.00	0.00	0.00	0.00	1.00
Stroke	319795.0	0.037740	0.190567	0.00	0.00	0.00	0.00	1.00
PhysicalHealth	319795.0	3.371710	7.950850	0.00	0.00	0.00	2.00	30.00
MentalHealth	319795.0	3.898366	7.955235	0.00	0.00	0.00	3.00	30.00
DiffWalking	319795.0	0.138870	0.345812	0.00	0.00	0.00	0.00	1.00
Sex	319795.0	0.475273	0.499389	0.00	0.00	0.00	1.00	1.00
PhysicalActivity	319795.0	0.775362	0.417344	0.00	1.00	1.00	1.00	1.00
SleepTime	319795.0	7.097075	1.436007	1.00	6.00	7.00	8.00	24.00
Asthma	319795.0	0.134061	0.340718	0.00	0.00	0.00	0.00	1.00
KidneyDisease	319795.0	0.036833	0.188352	0.00	0.00	0.00	0.00	1.00
SkinCancer	319795.0	0.093244	0.290775	0.00	0.00	0.00	0.00	1.00
AgeCategory_Category	319795.0	6.514536	3.564759	0.00	4.00	7.00	9.00	12.00
Diabetic_Category	319795.0	0.194002	0.496776	0.00	0.00	0.00	0.00	3.00
Race_Category	319795.0	1.554990	1.203594	1.00	1.00	1.00	1.00	6.00
GenHealth_Category	319795.0	2.175753	1.133848	1.00	1.00	2.00	3.00	5.00

12. Correlation matrix:

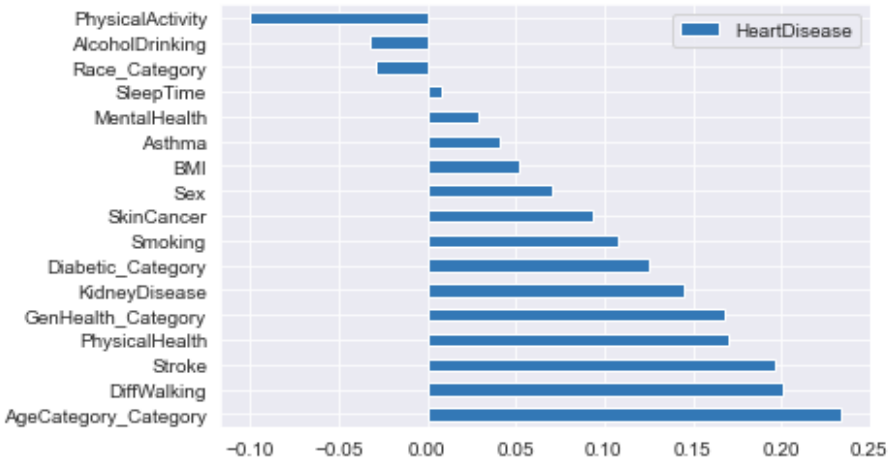
The correlation matrix is plotted against heart disease and the graph for the same is shown below.

From the below graphs, it is clear that the Physical Activity, Alcohol consumption, Race, Sleep Time do not contribute much to heart disease. Hence, these columns are removed in the pre-processing step as discussed before in this report.

In [186]:
```
data_corr = trans_df.corr()

corr_ft = plt.figure(figsize= (15, 15))
corr_ft = sea.heatmap(data_corr,annot=True)
plt.show()
```



Out[187]: <AxesSubplot:>



Inference Summary from EDA:

1. The data set that is chosen has a smaller number of people who have a heart disease.
2. People who have higher BMI are more prone to have heart disease especially if the BMI is above 25.5.
3. The older the people (above 65 years), the more they are susceptible to get a heart disease.
4. From the analysis above, it came out as drinking alcohol does not have an impact on heart disease.
5. Diabetic people and those above the age of 60, have more chances of suffering from a heart disease.
6. People who smoke are more prone to get a heart disease.
7. On comparing male to female ratio, the BMI of females is less than that of males. Also, the Mental Health, Physical Activity of females are slightly higher than that of males. Since those contribute to heart disease, females are less susceptible towards getting a heart disease when compared to males.
8. People who suffer from kidney disease have a high risk of getting heart diseases.
9. Those who are suffering from skin cancer have high chances of getting heart diseases.
10. Stroke is highly associated with heart disease, and people who had stroke in the past are prone to get heart disease
11. From the correlation matrix, it is seen that Physical activity, Alcohol drinking, race and sleep time doesn't influence the heart disease.

References:

- https://www.cdc.gov/nchs/fastats/heart-disease.htm
- https://pandas.pydata.org/docs/user_guide/index.html
- https://pandas.pydata.org/docs/reference/index.html
- https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.html
- https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease
- https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset
- https://www.geeksforgeeks.org/graph-plotting-in-python-set-1/
- https://stackoverflow.com/questions/8575062/how-to-show-matplotlib-plots
- https://stackoverflow.com/questions/53997862/pandas-groupby-two-columns-and-plot