Data Preparation

In [1]:
```
!wget https://raw.githubusercontent.com/keerthy456/Machine-Learning-Final-Proj
ect-Vakkalagadda-Keerthi/main/heart_disease.csv
```

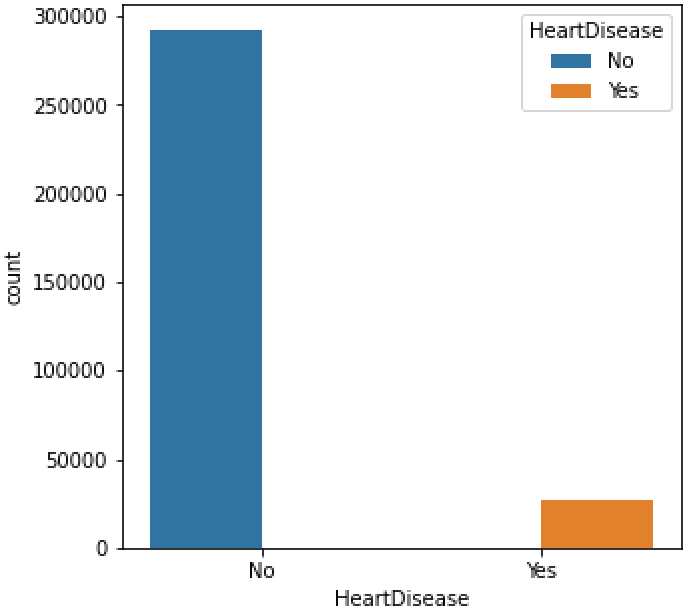
```
--2022-05-06 04:10:42--  https://raw.githubusercontent.com/keerthy456/Machine
-Learning-Final-Project-Vakkalagadda-Keerthi/main/heart_disease.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.11
1.133, 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.1
11.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25189554 (24M) [text/plain]
Saving to: ‘heart_disease.csv.2’

heart_disease.csv.2 100%[===================>]  24.02M   118MB/s    in 0.2s

2022-05-06 04:10:44 (118 MB/s) - ‘heart_disease.csv.2’ saved [25189554/251895
54]
```

In [3]:
```
import numpy as np
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split


%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from sklearn.metrics import confusion_matrix


from sklearn.linear_model import LinearRegression ,LogisticRegression
from sklearn.tree import DecisionTreeClassifier
```

In [4]:
```
heart_df = pd.read_csv('heart_disease.csv')
```

In [5]:
```
plt.figure(figsize = (5,5))
sns.countplot(x = heart_df['HeartDisease'], hue = 'HeartDisease', data = heart
_df)
```

Out[5]:
```
<AxesSubplot:xlabel='HeartDisease', ylabel='count'>
```



Sampling Dataset

In [6]:
```
class_no = heart_df[heart_df['HeartDisease'] == 'No']
class_yes = heart_df[heart_df['HeartDisease'] == 'Yes']
```

In [130]:
```
len(class_no)
```

Out[130]:
```
292422
```

In [7]:
```
class_no = class_no.sample(len(class_yes),replace=False)
new_df = pd.concat([class_no, class_yes], axis=0)
print('Target class Distibution after Sampling in :')
print(new_df['HeartDisease'].value_counts())
heart_df = new_df.copy()
```

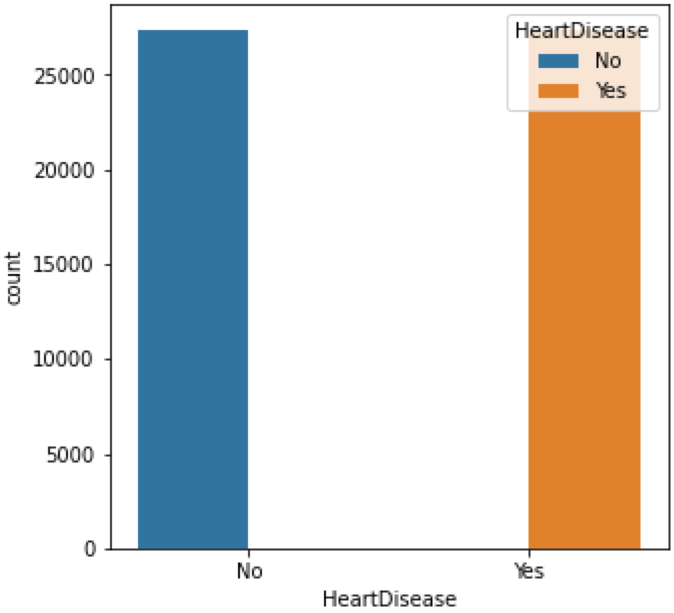
```
Target class Distibution after Sampling in :
No     27373
Yes    27373
Name: HeartDisease, dtype: int64
```

In [172]:

```
plt.figure(figsize = (5,5))
sns.countplot(x = new_df['HeartDisease'], hue = 'HeartDisease', data = new_df)
```

Out[172]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f5f6befde90>
```



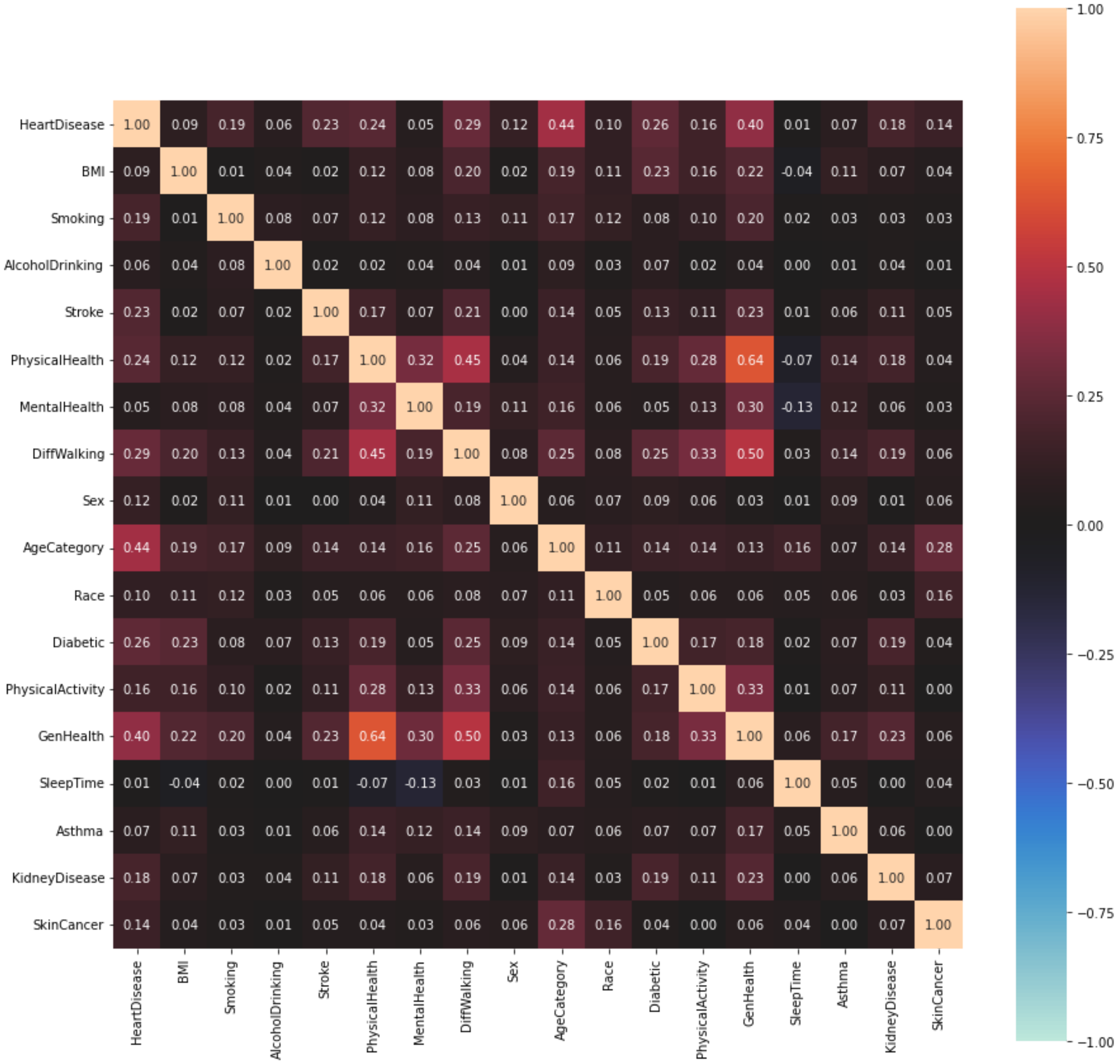
In [210]:

```
pip install dython
```

```
Requirement already satisfied: dython in /usr/local/lib/python3.7/dist-packag
es (0.7.1.post3)
Requirement already satisfied: pandas>=1.3.2 in /usr/local/lib/python3.7/dist
-packages (from dython) (1.3.5)
Requirement already satisfied: scipy>=1.7.1 in /usr/local/lib/python3.7/dist-
packages (from dython) (1.7.3)
Requirement already satisfied: matplotlib>=3.4.3 in /usr/local/lib/python3.7/
dist-packages (from dython) (3.5.2)
Requirement already satisfied: scikit-plot>=0.3.7 in /usr/local/lib/python3.
7/dist-packages (from dython) (0.3.7)
Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.7/di
st-packages (from dython) (0.11.2)
Requirement already satisfied: scikit-learn>=0.24.2 in /usr/local/lib/python
3.7/dist-packages (from dython) (1.0.2)
Requirement already satisfied: numpy>=1.19.5 in /usr/local/lib/python3.7/dist
-packages (from dython) (1.21.6)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python
3.7/dist-packages (from matplotlib>=3.4.3->dython) (2.8.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-
packages (from matplotlib>=3.4.3->dython) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in /usr/local/lib/python3.7/d
ist-packages (from matplotlib>=3.4.3->dython) (3.0.8)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.7/dist
-packages (from matplotlib>=3.4.3->dython) (7.1.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/
dist-packages (from matplotlib>=3.4.3->dython) (1.4.2)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.7/
dist-packages (from matplotlib>=3.4.3->dython) (4.33.3)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/di
st-packages (from matplotlib>=3.4.3->dython) (21.3)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/
dist-packages (from kiwisolver>=1.0.1->matplotlib>=3.4.3->dython) (4.2.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-
packages (from pandas>=1.3.2->dython) (2022.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-pack
ages (from python-dateutil>=2.7->matplotlib>=3.4.3->dython) (1.15.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python
3.7/dist-packages (from scikit-learn>=0.24.2->dython) (3.1.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-
packages (from scikit-learn>=0.24.2->dython) (1.1.0)
```

In [215]:

```
from dython.nominal import associations
associations(heart_df, figsize=(15,15))
```


HeartDisease 1.00 0.09 0.19 0.06 0.23 0.24 0.05 0.29 0.12 0.44 0.10 0.26 0.16 0.40 0.01 0.07 0.18 0.14
BMI 0.09 1.00 0.01 0.04 0.02 0.12 0.08 0.20 0.02 0.19 0.11 0.23 0.16 0.22 -0.04 0.11 0.07 0.04
Smoking 0.19 0.01 1.00 0.08 0.07 0.12 0.08 0.13 0.11 0.17 0.12 0.08 0.10 0.20 0.02 0.03 0.03 0.03
AlcoholDrinking 0.06 0.04 0.08 1.00 0.02 0.02 0.04 0.04 0.01 0.09 0.03 0.07 0.02 0.04 0.00 0.01 0.04 0.01
Stroke 0.23 0.02 0.07 0.02 1.00 0.17 0.07 0.21 0.00 0.14 0.05 0.13 0.11 0.23 0.01 0.06 0.11 0.05
PhysicalHealth 0.24 0.12 0.12 0.02 0.17 1.00 0.32 0.45 0.04 0.14 0.06 0.19 0.28 0.64 -0.07 0.14 0.18 0.04
MentalHealth 0.05 0.08 0.08 0.04 0.07 0.32 1.00 0.19 0.11 0.16 0.06 0.05 0.13 0.30 -0.13 0.12 0.06 0.03
DiffWalking 0.29 0.20 0.13 0.04 0.21 0.45 0.19 1.00 0.08 0.25 0.08 0.25 0.33 0.50 0.03 0.14 0.19 0.06
Sex 0.12 0.02 0.11 0.01 0.00 0.04 0.11 0.08 1.00 0.06 0.07 0.09 0.06 0.03 0.01 0.09 0.01 0.06
AgeCategory 0.44 0.19 0.17 0.09 0.14 0.14 0.16 0.25 0.06 1.00 0.11 0.14 0.14 0.13 0.16 0.07 0.14 0.28
Race 0.10 0.11 0.12 0.03 0.05 0.06 0.06 0.08 0.07 0.11 1.00 0.05 0.06 0.06 0.05 0.06 0.03 0.16
Diabetic 0.26 0.23 0.08 0.07 0.13 0.19 0.05 0.25 0.09 0.14 0.05 1.00 0.17 0.18 0.02 0.07 0.19 0.04
PhysicalActivity 0.16 0.16 0.10 0.02 0.11 0.28 0.13 0.33 0.06 0.14 0.06 0.17 1.00 0.33 0.01 0.07 0.11 0.00
GenHealth 0.40 0.22 0.20 0.04 0.23 0.64 0.30 0.50 0.03 0.13 0.06 0.18 0.33 1.00 0.06 0.17 0.23 0.06
SleepTime 0.01 -0.04 0.02 0.00 0.01 -0.07 -0.13 0.03 0.01 0.16 0.05 0.02 0.01 0.06 1.00 0.05 0.00 0.04
Asthma 0.07 0.11 0.03 0.01 0.06 0.14 0.12 0.14 0.09 0.07 0.06 0.07 0.07 0.17 0.05 1.00 0.06 0.00
KidneyDisease 0.18 0.07 0.03 0.04 0.11 0.18 0.06 0.19 0.01 0.14 0.03 0.19 0.11 0.23 0.00 0.06 1.00 0.07
SkinCancer 0.14 0.04 0.03 0.01 0.05 0.04 0.03 0.06 0.06 0.28 0.16 0.04 0.00 0.06 0.04 0.00 0.07 1.00
HeartDisease
BMI
Smoking
AlcoholDrinking
Stroke
PhysicalHealth
MentalHealth
DiffWalking
Sex
AgeCategory
Race
Diabetic
PhysicalActivity
GenHealth
SleepTime
Asthma
KidneyDisease
SkinCancer
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Out[215]:

```
{'ax': <matplotlib.axes._subplots.AxesSubplot at 0x7f5f6b3c2c50>,
 'corr':                    HeartDisease       BMI   Smoking  AlcoholDrinking  
Stroke  \
 HeartDisease          1.000000  0.086123  0.186801         0.062757  0.22633
1
 BMI                   0.086123  1.000000  0.011327         0.035059  0.01952
9
 Smoking               0.186801  0.011327  1.000000         0.083308  0.07455
1
 AlcoholDrinking       0.062757  0.035059  0.083308         1.000000  0.01942
8
 Stroke                0.226331  0.019529  0.074551         0.019428  1.00000
0
 PhysicalHealth        0.241695  0.119827  0.124253         0.024145  0.17045
4
 MentalHealth          0.047635  0.076881  0.083622         0.038517  0.07334
7
 DiffWalking           0.287334  0.197973  0.128690         0.042537  0.20580
7
 Sex                   0.123744  0.019856  0.111418         0.005145  0.00000
0
 AgeCategory           0.440379  0.186255  0.165518         0.090602  0.14229
1
 Race                  0.103719  0.105677  0.115258         0.028900  0.05316
6
 Diabetic              0.264801  0.234813  0.077663         0.073200  0.12789
5
 PhysicalActivity      0.163729  0.162181  0.104091         0.015783  0.10712
7
 GenHealth             0.398104  0.218810  0.200787         0.042866  0.22529
1
 SleepTime             0.014021 -0.043605  0.023065         0.002619  0.00869
8
 Asthma                0.072932  0.110901  0.030711         0.009336  0.05915
0
 KidneyDisease         0.181635  0.072841  0.034510         0.036570  0.11055
8
 SkinCancer            0.143247  0.040863  0.027861         0.010678  0.04682
6

                   PhysicalHealth  MentalHealth  DiffWalking       Sex  \
 HeartDisease            0.241695      0.047635     0.287334  0.123744   
 BMI                     0.119827      0.076881     0.197973  0.019856   
 Smoking                 0.124253      0.083622     0.128690  0.111418   
 AlcoholDrinking         0.024145      0.038517     0.042537  0.005145   
 Stroke                  0.170454      0.073347     0.205807  0.000000   
 PhysicalHealth          1.000000      0.324227     0.453900  0.042898   
 MentalHealth            0.324227      1.000000     0.194129  0.110779   
 DiffWalking             0.453900      0.194129     1.000000  0.081818   
 Sex                     0.042898      0.110779     0.081818  1.000000   
 AgeCategory             0.143431      0.157536     0.253090  0.057654   
 Race                    0.056797      0.061992     0.075858  0.067554   
 Diabetic                0.192377      0.054333     0.248460  0.091042   
 PhysicalActivity        0.276534      0.126524     0.325859  0.063045   
 GenHealth               0.640061      0.298198     0.496175  0.029681   
 SleepTime              -0.070978     -0.126860     0.033283  0.005607   
 Asthma                  0.142356      0.123548     0.144099  0.087855   
 KidneyDisease          0.176063      0.057116     0.190639  0.009989
 SkinCancer             0.040202      0.032021     0.063370  0.056208

                   AgeCategory      Race  Diabetic  PhysicalActivity  \
 HeartDisease         0.440379  0.103719  0.264801          0.163729
 BMI                  0.186255  0.105677  0.234813          0.162181
 Smoking              0.165518  0.115258  0.077663          0.104091
 AlcoholDrinking      0.090602  0.028900  0.073200          0.015783
 Stroke               0.142291  0.053166  0.127895          0.107127
 PhysicalHealth       0.143431  0.056797  0.192377          0.276534
 MentalHealth         0.157536  0.061992  0.054333          0.126524
 DiffWalking          0.253090  0.075858  0.248460          0.325859
 Sex                  0.057654  0.067554  0.091042          0.063045
 AgeCategory          1.000000  0.106135  0.137154          0.144289
 Race                 0.106135  1.000000  0.045507          0.056471
 Diabetic             0.137154  0.045507  1.000000          0.171012
 PhysicalActivity     0.144289  0.056471  0.171012          1.000000
 GenHealth            0.128188  0.056113  0.181264          0.330934
 SleepTime            0.157892  0.053026  0.017061          0.006354
 Asthma               0.065954  0.060598  0.072943          0.069547
 KidneyDisease        0.138081  0.033205  0.192191          0.114242
 SkinCancer           0.280373  0.160454  0.035720          0.000000

                   GenHealth  SleepTime    Asthma  KidneyDisease  SkinCancer
 HeartDisease       0.398104   0.014021  0.072932       0.181635    0.143247
 BMI                0.218810  -0.043605  0.110901       0.072841    0.040863
 Smoking            0.200787   0.023065  0.030711       0.034510    0.027861
 AlcoholDrinking    0.042866   0.002619  0.009336       0.036570    0.010678
 Stroke             0.225291   0.008698  0.059150       0.110558    0.046826
 PhysicalHealth     0.640061  -0.070978  0.142356       0.176063    0.040202
 MentalHealth       0.298198  -0.126860  0.123548       0.057116    0.032021
 DiffWalking        0.496175   0.033283  0.144099       0.190639    0.063370
 Sex                0.029681   0.005607  0.087855       0.009989    0.056208
 AgeCategory        0.128188   0.157892  0.065954       0.138081    0.280373
 Race               0.056113   0.053026  0.060598       0.033205    0.160454
 Diabetic           0.181264   0.017061  0.072943       0.192191    0.035720
 PhysicalActivity   0.330934   0.006354  0.069547       0.114242    0.000000
 GenHealth          1.000000   0.064136  0.173484       0.232674    0.059565
 SleepTime          0.064136   1.000000  0.054742       0.000934    0.039736
 Asthma             0.173484   0.054742  1.000000       0.059647    0.000000
 KidneyDisease      0.232674   0.000934  0.059647       1.000000    0.067498
 SkinCancer         0.059565   0.039736  0.000000       0.067498    1.000000
}
```

By looking at the above coorelation matrix I believe 'Alcohol Drinking', 'Mental Health', 'Sleep Time', 'Race' doesnot seem to be highly coorelated (individually/when combined with other features as well) with 'Target Feature - Heart Disease'. So I'll be dropping them from my dataset and train the model with remaining features.

In [8]:
```
numeric_features = heart_df.select_dtypes(include=[np.number])
```

In [134]:
```
categorical_features= [col for col in heart_df.columns if heart_df[col].dtypes
== 'object']
```

In [124]:
```
heart_df.isnull().any()
```

Out[124]:
```
HeartDisease        False
BMI                 False
Smoking             False
AlcoholDrinking     False
Stroke              False
PhysicalHealth      False
MentalHealth        False
DiffWalking         False
Sex                 False
AgeCategory         False
Race                False
Diabetic            False
PhysicalActivity    False
GenHealth           False
SleepTime           False
Asthma              False
KidneyDisease       False
SkinCancer          False
dtype: bool
```

In [9]:
```
heart_df = heart_df.drop(columns=['AlcoholDrinking', 'SleepTime', 'MentalHealth', 'Race'])
```

In [10]:
```
heart_df['heartdisease_GenHealth'] = heart_df.groupby('GenHealth')['HeartDisease'].transform('count')
heart_df['mean_PhysicalHealth'] = heart_df.groupby('DiffWalking')['PhysicalHealth'].transform('mean')
heart_df['BMI_Std'] = heart_df.groupby('PhysicalActivity')['BMI'].transform('std')
```

In [11]:
```
heart_df.groupby('GenHealth').count()
```

Out[11]:

	HeartDisease	BMI	Smoking	Stroke	PhysicalHealth	DiffWalking	Sex	AgeCateg
GenHealth								
Excellent	7664	7664	7664	7664	7664	7664	7664	7
Fair	9716	9716	9716	9716	9716	9716	9716	9
Good	17304	17304	17304	17304	17304	17304	17304	17
Poor	4537	4537	4537	4537	4537	4537	4537	4
Very good	15525	15525	15525	15525	15525	15525	15525	15

In [12]:
```
heart_df.head()
```

Out[12]:

	HeartDisease	BMI	Smoking	Stroke	PhysicalHealth	DiffWalking	Sex	AgeCategory
249975	No	24.96	No	No	0.0	No	Female	65-6
311944	No	28.19	Yes	No	0.0	No	Female	65-6
63414	No	23.40	Yes	No	0.0	No	Male	70-7
175588	No	30.52	Yes	No	0.0	No	Male	70-7
144586	No	28.97	Yes	No	0.0	No	Male	50-5

In [13]:
```
print('\nCategorical Columns\n')
heart_df.select_dtypes(include=['O']).nunique()
```

```

Categorical Columns

```

Out[13]:
```
HeartDisease         2
Smoking              2
Stroke               2
DiffWalking          2
Sex                  2
AgeCategory         13
Diabetic             4
PhysicalActivity     2
GenHealth            5
Asthma               2
KidneyDisease        2
SkinCancer           2
dtype: int64
```

From the above output we can see some of the categorical features are binary class variables which has 2 uniques values -yes/no, and some have more than 2 class values. So, for encoding binary features I've used Label encoder for converting yes/no to either 1 or 0 and dummies technique is used for categorical columns which has more than two unique values. Coming to dropping the unnecessary columns, I did not drop any columns because the dataset does not have any unique identifiers/ patient name/ address/zip code information and based on my intuition everything seems important. However the age Category is in categorical(each value is specified as range) and I'm converting it into integer by taking the mean if the given range.

In [14]:
```
for col in ['HeartDisease', 'Smoking', 'Stroke', 'DiffWalking', 'Sex', 'Physic
alActivity', 'Asthma', 'KidneyDisease', 'SkinCancer']:
    if heart_df[col].dtype == 'O':
        le = LabelEncoder()
        heart_df[col] = le.fit_transform(heart_df[col])
```

In [15]:
```
heart_df.head()
```

Out[15]:

	HeartDisease	BMI	Smoking	Stroke	PhysicalHealth	DiffWalking	Sex	AgeCategory	I
249975	0	24.96	0	0	0.0	0	0	65-69	
311944	0	28.19	1	0	0.0	0	0	65-69	
63414	0	23.40	1	0	0.0	0	1	70-74	
175588	0	30.52	1	0	0.0	0	1	70-74	
144586	0	28.97	1	0	0.0	0	1	50-54	

In [16]:
```
categoricals = heart_df[['GenHealth','Diabetic']]
categoricals.head()
```

Out[16]:

	GenHealth	Diabetic
249975	Good	No
311944	Excellent	No
63414	Very good	No
175588	Good	Yes
144586	Good	No

In [17]:
```
cat_dummies = pd.get_dummies(categoricals, drop_first=True)
cat_dummies.head()
```

Out[17]:

	GenHealth_Fair	GenHealth_Good	GenHealth_Poor	GenHealth_Very good	Diabetic_No, borderline diabetes	Diabet
249975	0	1	0	0	0	
311944	0	0	0	0	0	
63414	0	0	0	1	0	
175588	0	1	0	0	0	
144586	0	1	0	0	0	

In [18]:
```
# Drop the redundant columns
heart_df.drop(list(categoricals.columns), axis=1, inplace=True)
# concat the heart and dummies data frames.
heart_df = pd.concat([heart_df, cat_dummies], axis=1)
heart_df.head()
```

Out[18]:

	HeartDisease	BMI	Smoking	Stroke	PhysicalHealth	DiffWalking	Sex	AgeCategory	I
249975	0	24.96	0	0	0.0	0	0	65-69	
311944	0	28.19	1	0	0.0	0	0	65-69	
63414	0	23.40	1	0	0.0	0	1	70-74	
175588	0	30.52	1	0	0.0	0	1	70-74	
144586	0	28.97	1	0	0.0	0	1	50-54	

5 rows × 22 columns

In [19]:
```
encode_AgeCategory = {'55-59':57, '80 or older':80, '65-69':67,
                      '75-79':77,'40-44':42,'70-74':72,'60-64':62,
                      '50-54':52,'45-49':47,'18-24':21,'35-39':37,
                      '30-34':32,'25-29':27}
heart_df['AgeCategory'] = heart_df['AgeCategory'].apply(lambda x: encode_AgeCa
tegory[x])
heart_df['AgeCategory'] = heart_df['AgeCategory'].astype('int64')
```

In [20]:
```
heart_df.head()
```

Out[20]:

	HeartDisease	BMI	Smoking	Stroke	PhysicalHealth	DiffWalking	Sex	AgeCategory	I
249975	0	24.96	0	0	0.0	0	0	67	
311944	0	28.19	1	0	0.0	0	0	67	
63414	0	23.40	1	0	0.0	0	1	72	
175588	0	30.52	1	0	0.0	0	1	72	
144586	0	28.97	1	0	0.0	0	1	52	

5 rows × 22 columns

Normalization/Scaling

The range of continuous features are different. Here, I am scaling them to be in-between 0 to 1 by dividing by the maximum value of the respective column

In [21]:
```python
from sklearn.preprocessing import StandardScaler
num_cols = ['BMI', 'PhysicalHealth']
scaler = StandardScaler()
heart_df[num_cols] = scaler.fit_transform(heart_df[num_cols])
```

In [22]:
```python
heart_df.describe()[1:][['BMI','PhysicalHealth']].T.style.background_gradient(
cmap='Blues')
```

Out[22]:

	mean	std	min	25%	50%	75%	max
BMI	0.000000	1.000009	-2.573955	-0.682142	-0.162125	0.495226	8.948204
PhysicalHealth	-0.000000	1.000009	-0.538968	-0.538968	-0.538968	-0.036554	2.475512

In [23]:
```python
heart_df = heart_df.drop(columns = ['heartdisease_GenHealth', 'mean_PhysicalHe
alth','BMI_Std',])
```

In [24]:
```python
#Select Features
features = heart_df.drop(columns =['HeartDisease'], axis = 1)
#Select Target
target = heart_df['HeartDisease']
# Set Training and Testing Data
from sklearn.model_selection import train_test_split
```

In [25]:
```python
high = len(heart_df['HeartDisease']) - sum(heart_df['HeartDisease'])
print("Baseline accuracy : ",high/len(heart_df['HeartDisease']))
```

```
Baseline accuracy :  0.5
```

In [26]:
```python
from sklearn import metrics
```

Linear Regression

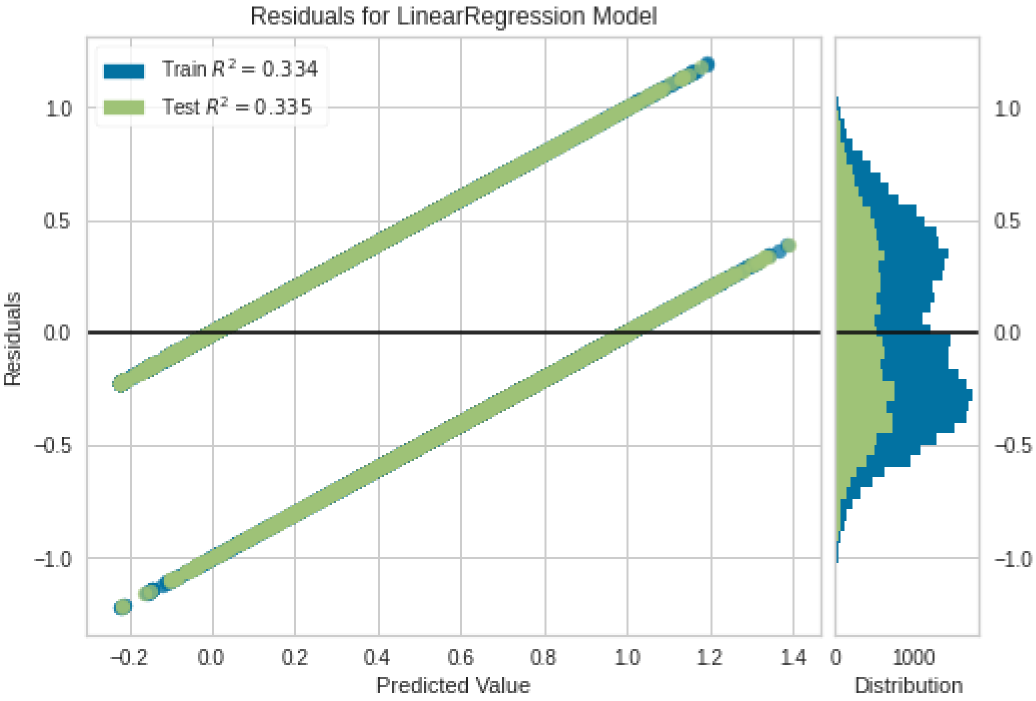
In [62]:
```python
X_train, X_test, y_train, y_test = train_test_split(features, target, test_siz
e = 0.1, random_state=33)
```

In [61]:
```python
# plot impact of logloss for single forecasts
from sklearn.metrics import log_loss
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
lin_reg_pred = lin_reg.predict(X_test)
score = lin_reg.score(X_test,y_test)

print("Linear Regression Accuracy",score)
```

```
Linear Regression Accuracy 0.3319457969522649
```

In [30]:
```
from yellowbrick.regressor import ResidualsPlot
visual = ResidualsPlot(lin_reg)
visual.fit(X_train,y_train)
visual.score(X_test,y_test)
visual.poof()
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X do
es not have valid feature names, but LinearRegression was fitted with feature
names
  "X does not have valid feature names, but"
```



Out[30]:
```
<AxesSubplot:title={'center':'Residuals for LinearRegression Model'}, xlabel
='Predicted Value', ylabel='Residuals'>
```

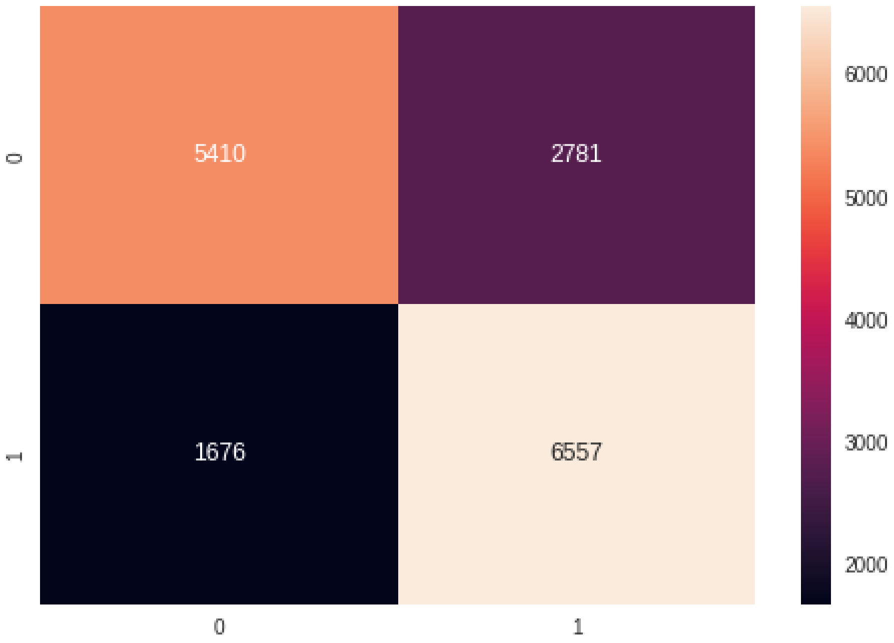
Decision Tree

In [63]:
```
dt_clf = DecisionTreeClassifier(max_depth=4)
dt_clf.fit(X_train, y_train)
dt_pred = dt_clf.predict(X_test)
dt_acc = metrics.accuracy_score(y_test, dt_pred)
print("Decision Tree Accuracy",100*dt_acc)
```

```
Decision Tree Accuracy 72.94977168949772
```

In [53]:

```
con_matrix = confusion_matrix(y_test,dt_pred)
sns.heatmap(con_matrix,annot=True,fmt="d")
plt.show()
```



In [ ]:

```
```

In [55]:

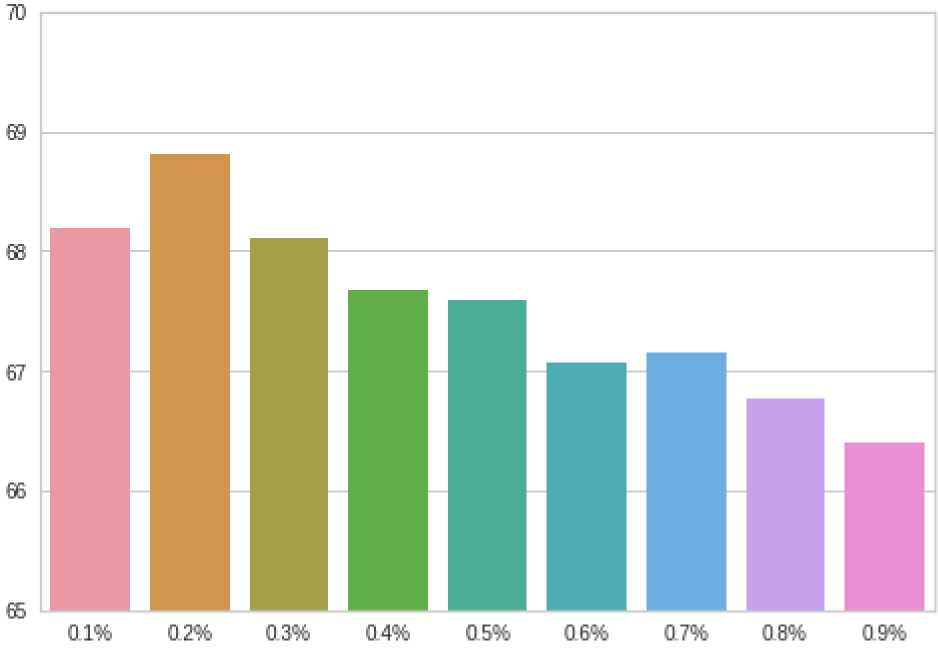
```
size = 0.1
accuracy_list=[]
dataset_ratio = []
for i in range(1,10):
  print('Test set size: ',(size))
  index = int((i/10)*len(new_df['HeartDisease']))
  X_train, X_test, y_train, y_test = train_test_split(features, target, test_size = (i/10), random_state=100)
  dt_clf1 = DecisionTreeClassifier()
  dt_clf1.fit(X_train, y_train)
  dt_pred = dt_clf1.predict(X_test)
  dt_acc1 = metrics.accuracy_score(y_test, dt_pred)
  accuracy_list.append(100*dt_acc1)
  print("Accuracy: ",100*dt_acc1)
  size = round(size, 1)
  dataset_ratio.append(str(i/10)+'%')
  size = round((size+0.1), 1)
```

```
Test set size:  0.1
Accuracy:  68.18264840182648
Test set size:  0.2
Accuracy:  68.80365296803653
Test set size:  0.3
Accuracy:  68.11373599610326
Test set size:  0.4
Accuracy:  67.6743230284488
Test set size:  0.5
Accuracy:  67.58849961641033
Test set size:  0.6
Accuracy:  67.07866536775451
Test set size:  0.7
Accuracy:  67.15027529160035
Test set size:  0.8
Accuracy:  66.76713016873302
Test set size:  0.9
Accuracy:  66.39673648319533
```

do sampling

In [57]:
```
g = sns.barplot(x = dataset_ratio, y=accuracy_list)
g.set_ylim(65,70)
```

Out[57]:
```
(65.0, 70.0)
```

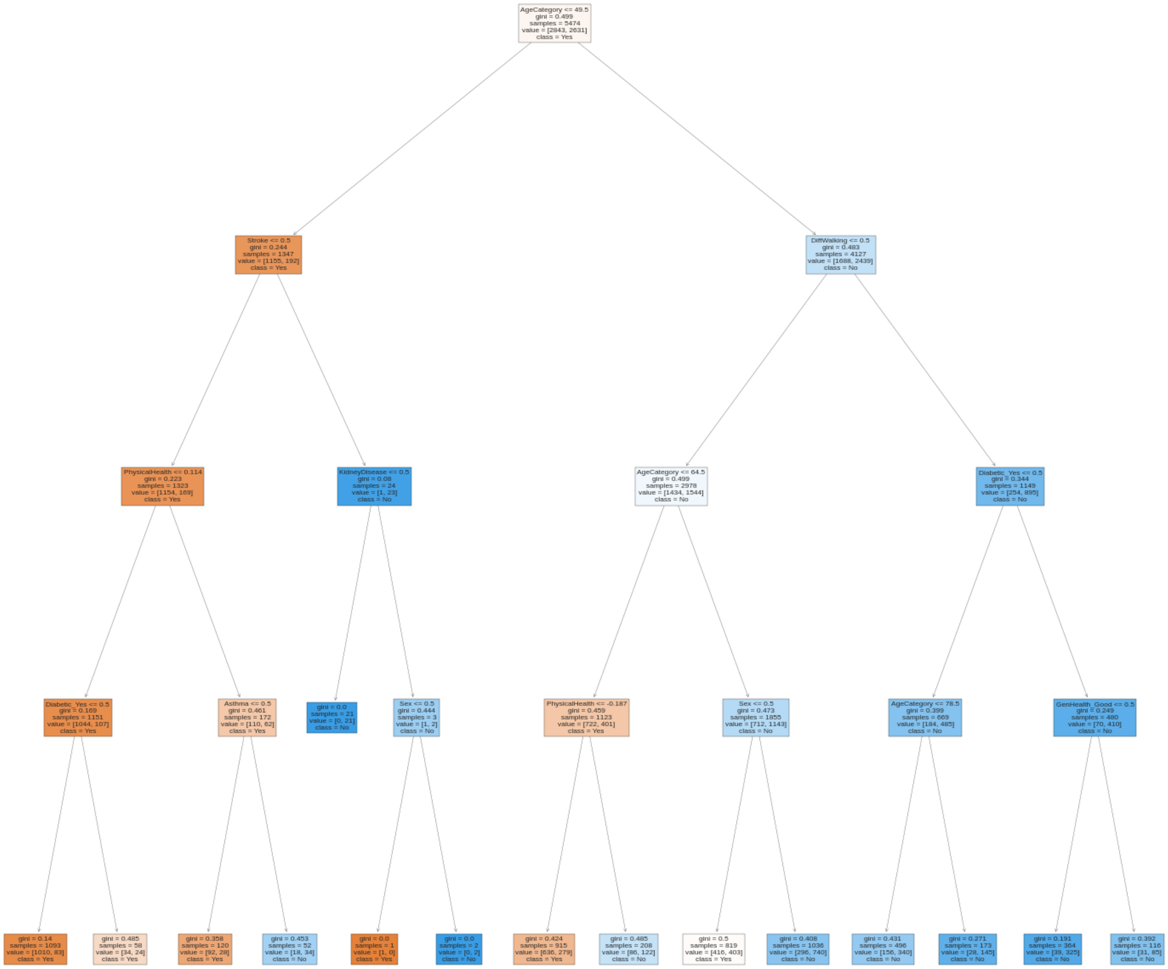


For our conclusion, this dataset does not suitable for classification in Heart Disease cause data overlapping, unbalanced data, low correlation and a lot of outliers.One single feature might not say much but combinations might say more than that! I will perform this in another notebook

In [45]:
```
from sklearn import tree
```

In [46]:

```
fig = plt.figure(figsize=(25,25))
v = tree.plot_tree(dt_clf,
                   feature_names=features.columns,
                   class_names=['Yes', 'No'],
                   filled=True)
```


AgeCategory <= 49.5 gini = 0.499 samples = 5474 value = [2843, 2631] class = Yes
Stroke <= 0.5 gini = 0.244 samples = 1347 value = [1155, 192] class = Yes
DiffWalking <= 0.5 gini = 0.483 samples = 4127 value = [1688, 2439] class = No
PhysicalHealth <= 0.114 gini = 0.223 samples = 1323 value = [1154, 169] class = Yes
KidneyDisease <= 0.5 gini = 0.08 samples = 24 value = [1, 23] class = No
AgeCategory <= 64.5 gini = 0.499 samples = 2978 value = [1434, 1544] class = No
Diabetic_Yes <= 0.5 gini = 0.344 samples = 1149 value = [254, 895] class = No
Diabetic_Yes <= 0.5 gini = 0.169 samples = 1151 value = [1044, 107] class = Yes
Asthma <= 0.5 gini = 0.461 samples = 172 value = [110, 62] class = Yes
gini = 0.0 samples = 21 value = [0, 21] class = No
Sex <= 0.5 gini = 0.444 samples = 3 value = [1, 2] class = No
PhysicalHealth <= -0.187 gini = 0.459 samples = 1123 value = [722, 401] class = Yes
Sex <= 0.5 gini = 0.473 samples = 1855 value = [712, 1143] class = No
AgeCategory <= 78.5 gini = 0.399 samples = 669 value = [184, 485] class = No
GenHealth_Good <= 0.5 gini = 0.249 samples = 480 value = [70, 410] class = No
gini = 0.14 samples = 1093 value = [1010, 83] class = Yes
gini = 0.485 samples = 58 value = [34, 24] class = Yes
gini = 0.358 samples = 120 value = [92, 28] class = Yes
gini = 0.453 samples = 52 value = [18, 34] class = No
gini = 0.0 samples = 1 value = [1, 0] class = Yes
gini = 0.0 samples = 2 value = [0, 2] class = No
gini = 0.424 samples = 915 value = [636, 279] class = Yes
gini = 0.485 samples = 208 value = [86, 122] class = No
gini = 0.5 samples = 819 value = [416, 403] class = Yes
gini = 0.408 samples = 1036 value = [296, 740] class = No
gini = 0.431 samples = 496 value = [156, 340] class = No
gini = 0.271 samples = 173 value = [28, 145] class = No
gini = 0.191 samples = 364 value = [39, 325] class = No
gini = 0.392 samples = 116 value = [31, 85] class = No