Stroke Prediction : A Machine Learning Approach

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import sklearn
from scipy.spatial.distance import pdist,squareform
from sklearn.metrics import pairwise_distances
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import warnings
warnings.filterwarnings("ignore")
from sklearn import neighbors, linear_model, svm, tree, ensemble
from sklearn.preprocessing import LabelEncoder
from pandas_profiling import ProfileReport
from sklearn.feature_selection import SelectKBest, f_classif
from scipy.spatial.distance import pdist,squareform
from sklearn.metrics import pairwise_distances
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from yellowbrick.cluster import SilhouetteVisualizer
sns.set()
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn import metrics,tree
from sklearn.tree import DecisionTreeClassifier,export_graphviz
from sklearn.naive_bayes import BernoulliNB
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
```

Importing Dataset

In [2]:
```
db=pd.read_csv("C:/Users/Dell/Downloads/healthcare-dataset-stroke-data.csv")
db.head(5)
```

Out[2]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	av
0	9046	Male	67.0	0	1	Yes	Private	Urban	
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	
2	31112	Male	80.0	0	1	Yes	Private	Rural	
3	60182	Female	49.0	0	0	Yes	Private	Urban	
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	

In [3]:
```
#checking for null values
db.isnull().sum()
```

Out[3]:
```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

In [4]:
```
#dropping null values
db=db.dropna()
db
```

Out[4]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
0	9046	Male	67.0	0	1	Yes	Private	Urban
2	31112	Male	80.0	0	1	Yes	Private	Rural
3	60182	Female	49.0	0	0	Yes	Private	Urban
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural
5	56669	Male	81.0	0	0	Yes	Private	Urban
...	...	...	...	...	...	...	...	...
5104	14180	Female	13.0	0	0	No	children	Rural
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural
5108	37544	Male	51.0	0	0	Yes	Private	Rural
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban

4909 rows × 12 columns

In [5]:
```
db.isnull().sum()
```

Out[5]:
```
id                 0
gender             0
age                0
hypertension       0
heart_disease      0
ever_married       0
work_type          0
Residence_type     0
avg_glucose_level  0
bmi                0
smoking_status     0
stroke             0
dtype: int64
```

In [6]:
```
#checking for duplicate values
db_dup = db.duplicated().any()
print(db_dup)
```

```
False
```

In [7]:
```
db
```

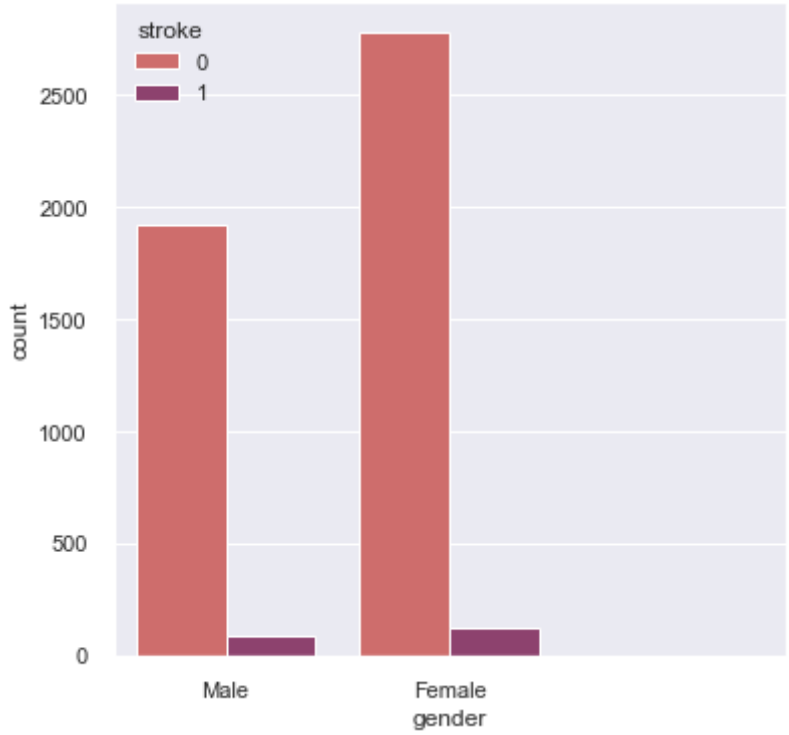
Out[7]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
0	9046	Male	67.0	0	1	Yes	Private	Urban
2	31112	Male	80.0	0	1	Yes	Private	Rural
3	60182	Female	49.0	0	0	Yes	Private	Urban
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural
5	56669	Male	81.0	0	0	Yes	Private	Urban
...	...	...	...	...	...	...	...	...
5104	14180	Female	13.0	0	0	No	children	Rural
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural
5108	37544	Male	51.0	0	0	Yes	Private	Rural
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban

4909 rows × 12 columns

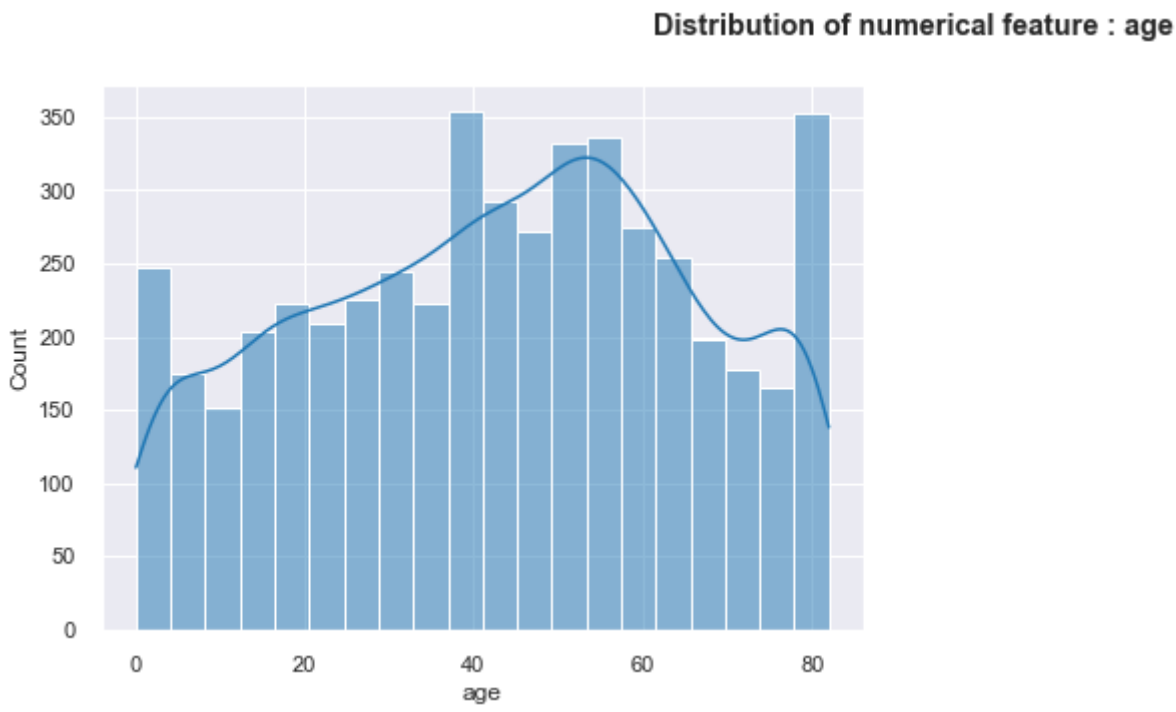
Visualizations

In [8]:
```
plt.figure(figsize=(6,6))
sns.countplot(x='gender',hue='stroke',data= db,palette="flare")
plt.xticks([1,0],['  Male  ','Female    '])
plt.show()
```

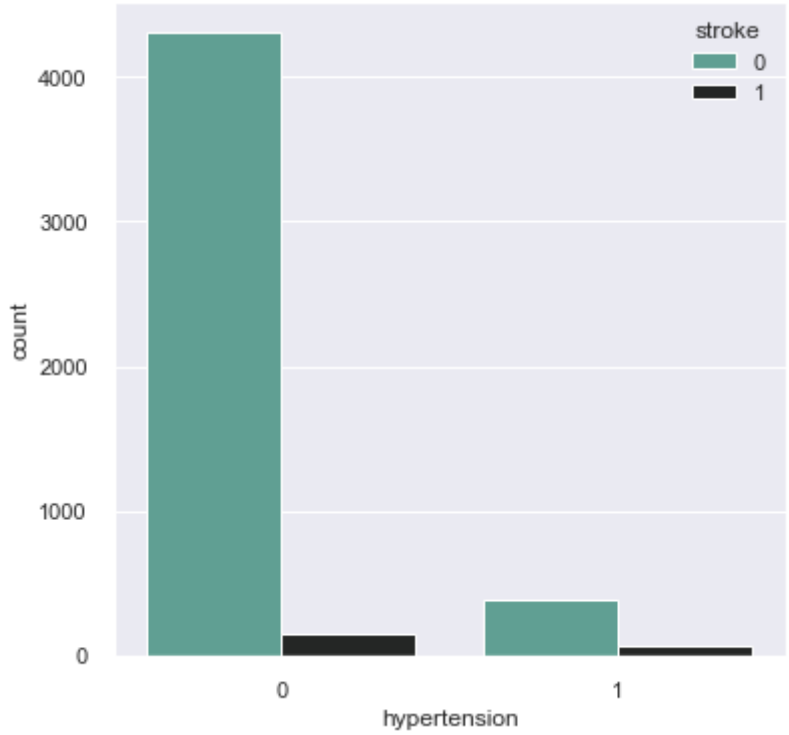


In [9]:
```
def continuous_plot(feature_name):
    plt.figure(figsize=(15,5))
    plt.suptitle('Distribution of numerical feature : {}'.format(feature_name), fo
    plt.style.use('seaborn-dark-palette')
    plt.subplot(1,2,1)
    sns.histplot(data=db,x=feature_name , kde=True, color= 'tab:blue')
```

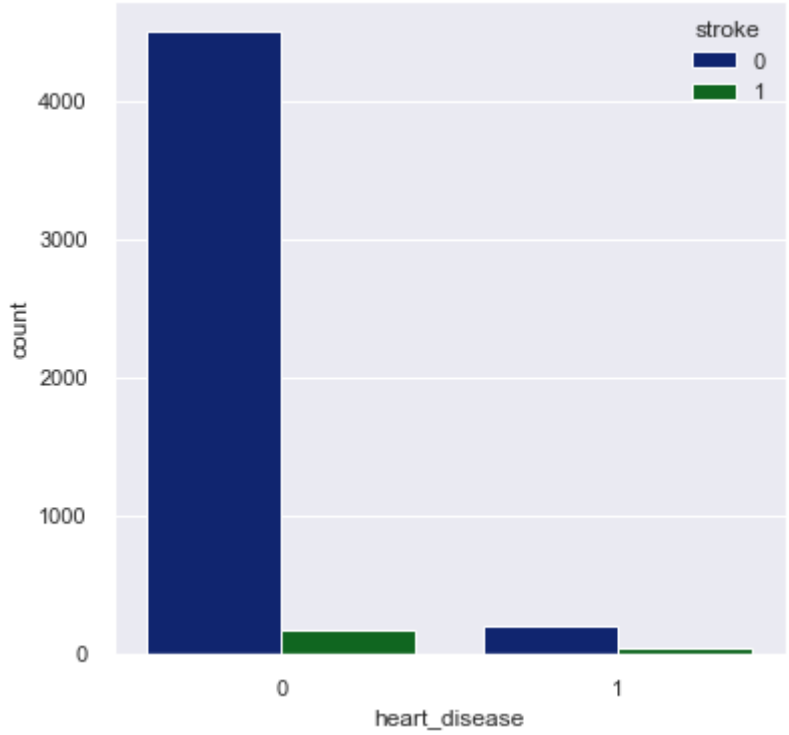
In [10]:
```
continuous_features=['age']
list(map(continuous_plot,continuous_features))
plt.show()
```



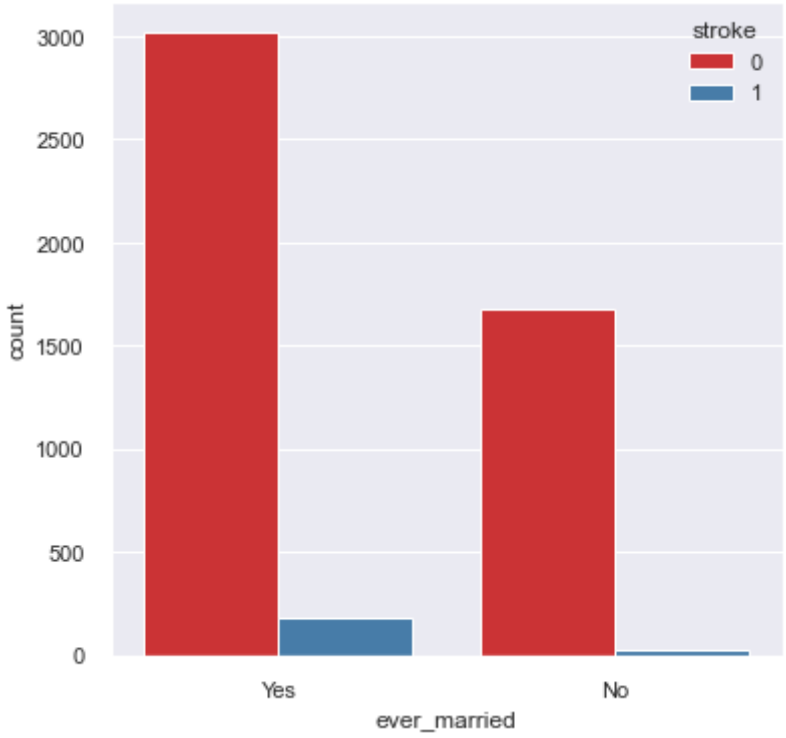
In [11]:
```
plt.figure(figsize=(6,6))
sns.countplot(x='hypertension',hue='stroke',data= db,palette="dark:#5A9_r")
plt.xticks([1,0]),[1,0]
plt.show()
```



In [12]:
```
plt.figure(figsize=(6,6))
sns.countplot(x='heart_disease',hue='stroke',data= db)
plt.xticks([1,0]),['1','0']
plt.show()
```



In [13]:
```
plt.figure(figsize=(6,6))
sns.countplot(x='ever_married',hue='stroke',data= db,palette="Set1")
plt.xticks([1,0]),['  Yes   ','No    ']
plt.show()
```



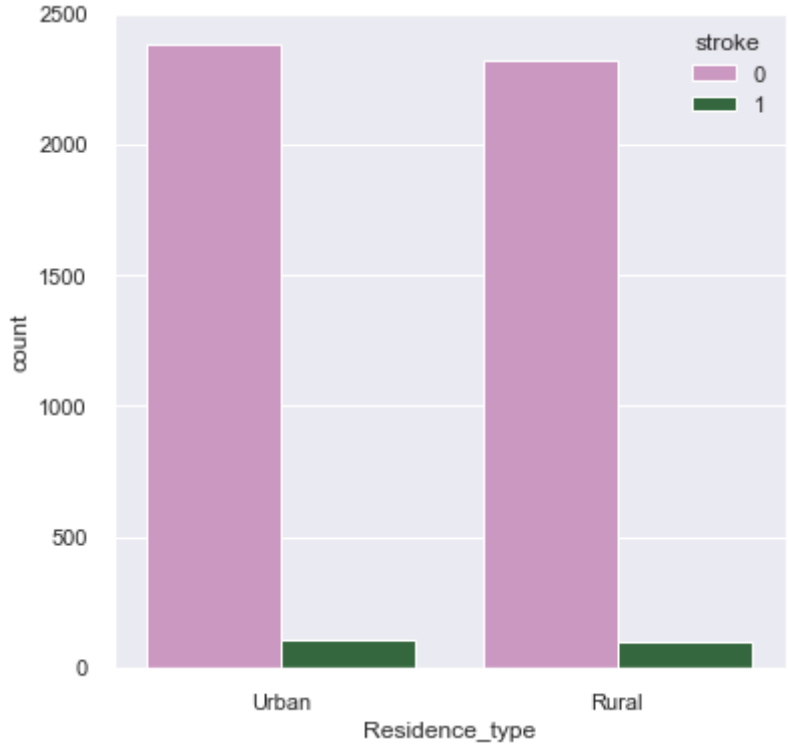
In [14]:
```
db
```

Out[14]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
0	9046	Male	67.0	0	1	Yes	Private	Urban
2	31112	Male	80.0	0	1	Yes	Private	Rural
3	60182	Female	49.0	0	0	Yes	Private	Urban
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural
5	56669	Male	81.0	0	0	Yes	Private	Urban
...	...	...	...	...	...	...	...	...
5104	14180	Female	13.0	0	0	No	children	Rural
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural
5108	37544	Male	51.0	0	0	Yes	Private	Rural
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban

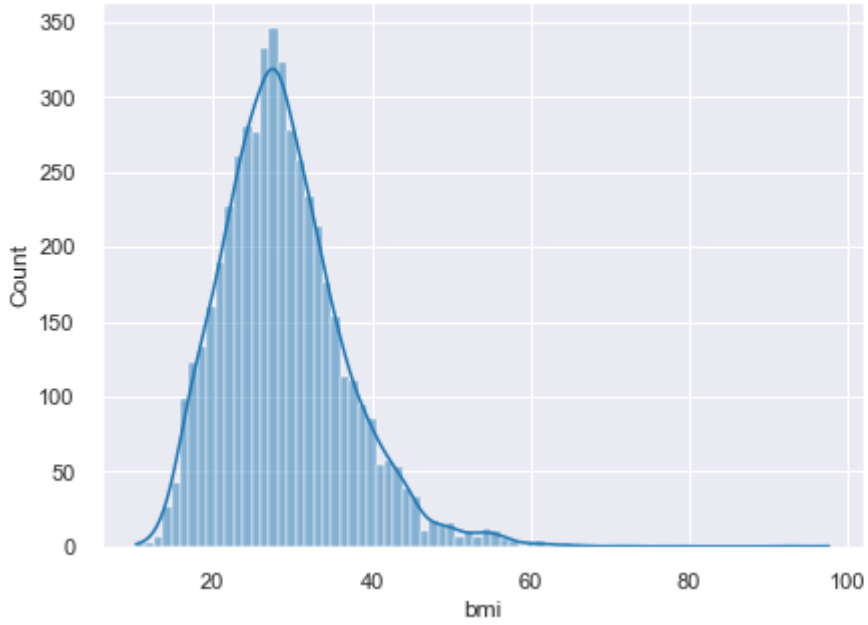
4909 rows × 12 columns

In [15]:
```
plt.figure(figsize=(6,6))
sns.countplot(x='Residence_type',hue='stroke',data= db,palette='cubehelix_r')
plt.xticks([1,0]),['  Urban  ','Rural    ']
plt.show()
```

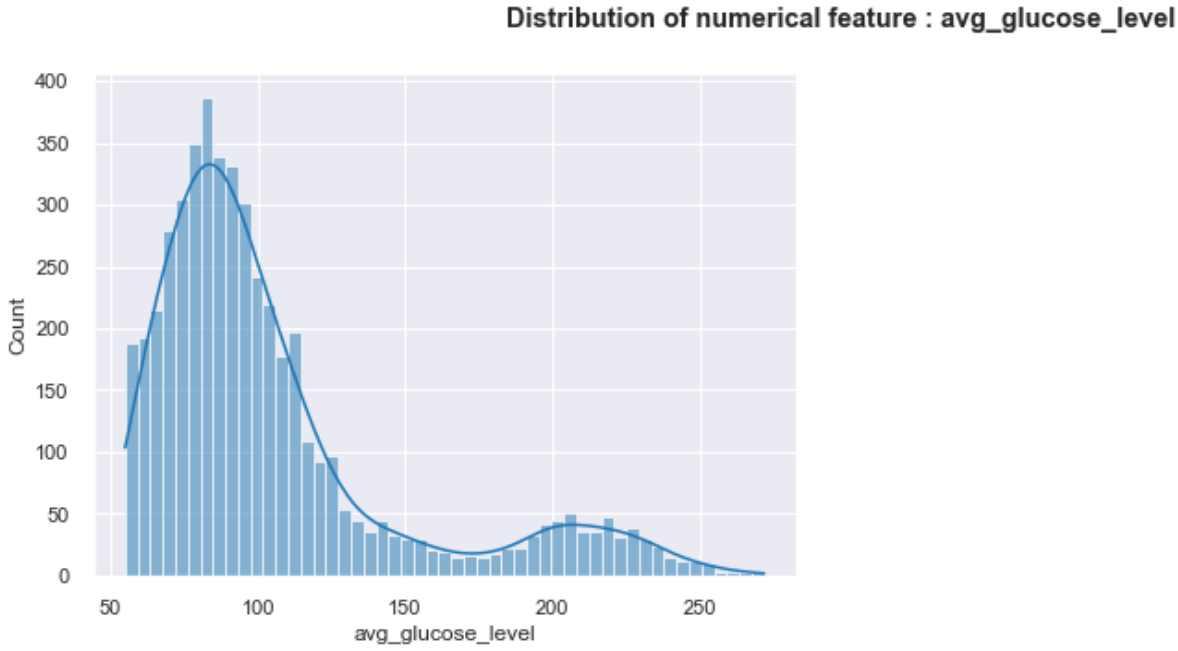


In [16]:
```
continuous_features=['bmi']
list(map(continuous_plot,continuous_features))
plt.show()
```

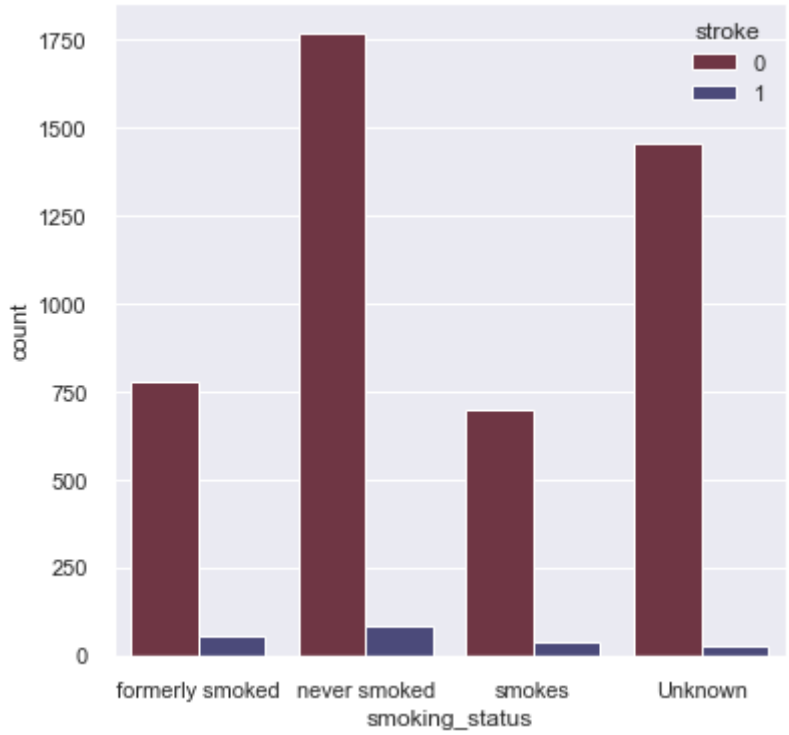
Distribution of numerical feature : bmi



In [17]:
```
continuous_features=['avg_glucose_level']
list(map(continuous_plot,continuous_features))
plt.show()
```



In [18]:
```
plt.figure(figsize=(6,6))
sns.countplot(x='smoking_status',hue='stroke',data= db,palette="icefire_r")
plt.xticks([3,2,1,0]),['  formerly smoked  ','never smoked','smokes','Unknown']
plt.show()
```

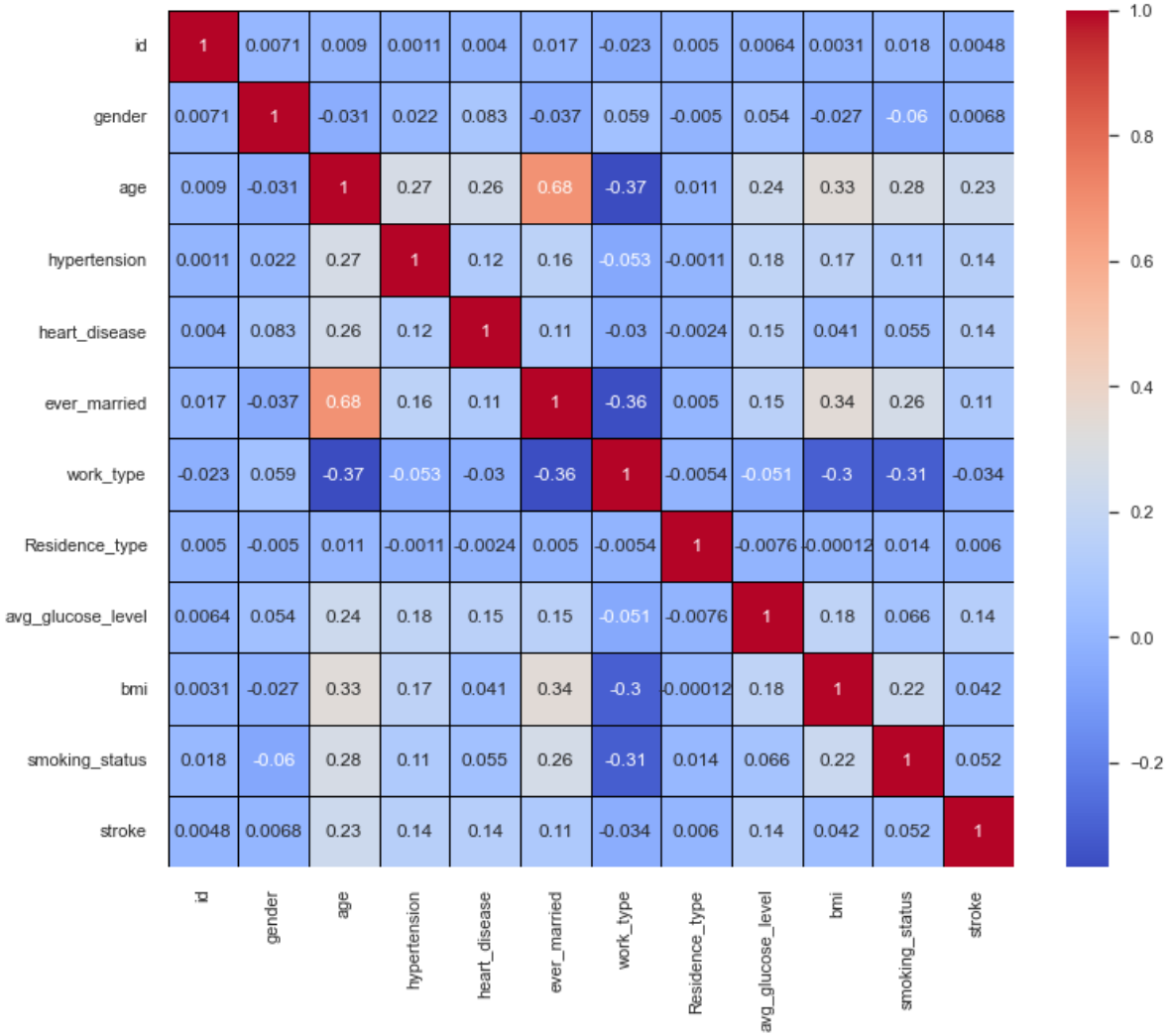


Label Encoding

In [19]:
```
LE_gender = LabelEncoder()
db['gender'] = LE_gender.fit_transform(db['gender'])
LE_ever_married = LabelEncoder()
db['ever_married'] = LE_gender.fit_transform(db['ever_married'])
LE_work_type = LabelEncoder()
db['work_type'] = LE_gender.fit_transform(db['work_type'])
LE_Residence_type = LabelEncoder()
db['Residence_type'] = LE_gender.fit_transform(db['Residence_type'])
LE_smoking_status = LabelEncoder()
db['smoking_status'] = LE_gender.fit_transform(db['smoking_status'])
```

Feature Selection

In [20]:
```
#Pearson Correlation Heat Map
plt.figure(figsize=(12,10))
cor = db.corr()
sns.heatmap(cor, annot=True, cmap='coolwarm',linewidths = 0.5,linecolor='black')
plt.show()
```



In [21]:
```
def correlation(ds, t):
    column_corr = set()
    corr_matrix = ds.corr()
    for n in range(len(corr_matrix.columns)):
        for m in range(n):
            if abs(corr_matrix.iloc[n, m]) > t:
                colname = corr_matrix.columns[n]
```

```
                    column_corr.add(colname)
    return column_corr
```

In [22]:
```
correlation_features = correlation(db,0.5)
len(set(correlation_features))
```

Out[22]:
```
1
```

In [23]:
```
correlation_features
```

Out[23]:
```
{'ever_married'}
```

In [24]:
```
X_db=db.drop(labels=['stroke'],axis=1)
y_db=db['stroke']
```

In [25]:
```
#K-Means for feature Selection
selector = SelectKBest(f_classif, k=8)
selector.fit(X_db, y_db)
np.set_printoptions(precision=3)
print(selector.scores_)
print(X_db.columns.values)
```

```
[1.149e-01 2.241e-01 2.800e+02 1.017e+02 9.518e+01 5.480e+01 5.598e+00
 1.785e-01 9.659e+01 8.826e+00 1.308e+01]
['id' 'gender' 'age' 'hypertension' 'heart_disease' 'ever_married'
 'work_type' 'Residence_type' 'avg_glucose_level' 'bmi' 'smoking_status']
```

In [26]:
```
feature_weights = dict(zip(X_db.columns.values, selector.scores_))
feature_weights
```

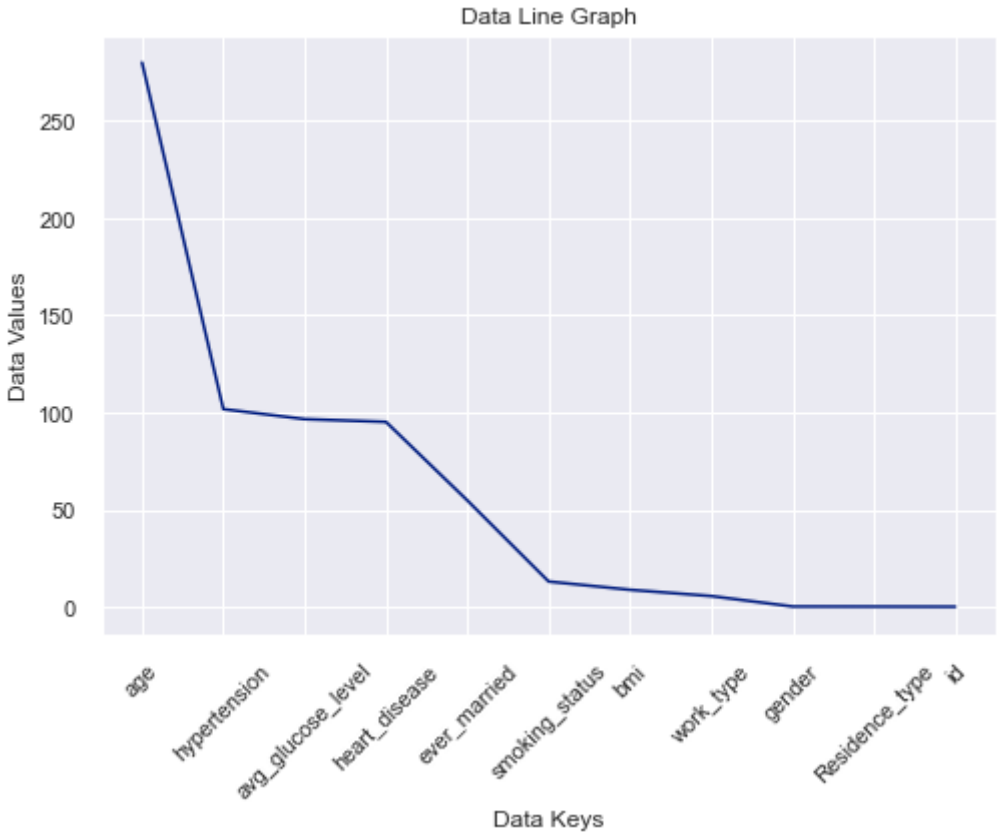
Out[26]:
```
{'id': 0.11490138441144958,
 'gender': 0.2240734239933916,
 'age': 279.9809183274611,
 'hypertension': 101.7293613811758,
 'heart_disease': 95.1755602934637,
 'ever_married': 54.79673431008887,
 'work_type': 5.597754138338402,
 'Residence_type': 0.178513856272716,
 'avg_glucose_level': 96.58507202364312,
 'bmi': 8.826499990596135,
 'smoking_status': 13.078820608911627}
```

In [27]:
```
feature_weights = dict(sorted(feature_weights.items(), key=lambda item: item[1], r
feature_weights
```

Out[27]:
```
{'age': 279.9809183274611,
 'hypertension': 101.7293613811758,
 'avg_glucose_level': 96.58507202364312,
 'heart_disease': 95.1755602934637,
 'ever_married': 54.79673431008887,
 'smoking_status': 13.078820608911627,
 'bmi': 8.826499990596135,
 'work_type': 5.597754138338402,
 'gender': 0.2240734239933916,
 'Residence_type': 0.178513856272716,
 'id': 0.11490138441144958}
```

In [28]:
```
data = {'age': 279.9809183274611,
        'hypertension': 101.7293613811758,
        'avg_glucose_level': 96.58507202364312,
        'heart_disease': 95.1755602934637,
        'ever_married': 54.79673431008887,
        'smoking_status': 13.078820608911627,
```

```
        'bmi': 8.826499990596135,
        'work_type': 5.597754138338402,
        'gender': 0.2240734239933916,
        'Residence_type': 0.178513856272716,
        'id': 0.11490138441144958}

# Sort the data by the values in descending order
data_sorted = {k: v for k, v in sorted(data.items(), key=lambda item: item[1], rev

# Extract the keys and values as lists
keys = list(data_sorted.keys())
values = list(data_sorted.values())

# Plot the line graph
plt.plot(keys, values)

# Add title and axis labels
plt.title('Data Line Graph')
plt.xlabel('Data Keys')
plt.ylabel('Data Values')

# Rotate the x-axis labels by 45 degrees
plt.xticks(rotation=45)

# Show the graph
plt.show()
```



In [29]:
```
#Dropping Least Significant Labels
X_train_db=X_db.drop(labels=['id'],axis=1)
X_train_db
```

Out[29]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_gl
0	1	67.0	0	1	1	2	1	
2	1	80.0	0	1	1	2	0	
3	0	49.0	0	0	1	2	1	
4	0	79.0	1	0	1	3	0	
5	1	81.0	0	0	1	2	1	
...	...	...	...	...	...	...	...	
5104	0	13.0	0	0	0	4	0	
5106	0	81.0	0	0	1	3	1	
5107	0	35.0	0	0	1	3	0	
5108	1	51.0	0	0	1	2	0	
5109	0	44.0	0	0	1	0	1	

4909 rows × 10 columns

Normalization

In [30]:
```
#Normalization for Glucose Level
average='avg_glucose_level'
X_train_db[average]=(X_train_db[average] - X_train_db[average].min()) / (X_train_dl
```

In [31]:
```
X_train_db
```

Out[31]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_gl
0	1	67.0	0	1	1	2	1	
2	1	80.0	0	1	1	2	0	
3	0	49.0	0	0	1	2	1	
4	0	79.0	1	0	1	3	0	
5	1	81.0	0	0	1	2	1	
...	...	...	...	...	...	...	...	
5104	0	13.0	0	0	0	4	0	
5106	0	81.0	0	0	1	3	1	
5107	0	35.0	0	0	1	3	0	
5108	1	51.0	0	0	1	2	0	
5109	0	44.0	0	0	1	0	1	

4909 rows × 10 columns

In [32]:
```
#Normalization for BMI
average='bmi'
X_train_db[average]=(X_train_db[average] - X_train_db[average].min()) / (X_train_dl
```

In [33]:
```
#Normalization for Age
average='age'
X_train_db[average]=(X_train_db[average] - X_train_db[average].min()) / (X_train_db
```

In [34]:
```
X_train_db
```

Out[34]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	av
0	1	0.816895	0	1	1	2	1	
2	1	0.975586	0	1	1	2	0	
3	0	0.597168	0	0	1	2	1	
4	0	0.963379	1	0	1	3	0	
5	1	0.987793	0	0	1	2	1	
...	...	...	...	...	...	...	...	
5104	0	0.157715	0	0	0	4	0	
5106	0	0.987793	0	0	1	3	1	
5107	0	0.426270	0	0	1	3	0	
5108	1	0.621582	0	0	1	2	0	
5109	0	0.536133	0	0	1	0	1	

4909 rows × 10 columns

Pandas profiling

In [35]:
```
import pandas_profiling
from pandas_profiling import ProfileReport
```

In [36]:
```
profile = ProfileReport(X_train_db, title="Pandas Profile Report")
profile
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

Overview

Dataset statistics

Number of variables	10
Number of observations	4909
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	455.0 KiB
Average record size in memory	94.9 B

Variable types

Categorical	7
Numeric	3

Alerts

`age` is highly overall correlated with `ever_married` and 2 other fields (ever_married, work_type, smoking_status)	**High correlation**
`ever_married` is highly overall correlated with `age` and	**High correlation**

Out[36]:

Training and testing split

In [37]:
```
X1_train, X1_Val, Y1_train, Y1_Val = train_test_split(X_train_db, y_db, test_size=
```

Decision Tree

In [38]:
```
clf = DecisionTreeClassifier(criterion='entropy')
clf1 = clf.fit(X1_train,Y1_train)
```

```
Y1_pred = clf1.predict(X1_Val)
```

In [39]:
```
fn=X_train_db.keys()
cn=['no','yes']
plt.figure(figsize=(15,15))
tree.plot_tree(clf1,feature_names = fn,class_names=cn,filled=True)
```

Out[39]:
```
[Text(0.2953392094017094, 0.9772727272727273, 'age <= 0.652\nentropy = 0.251\nsamp
les = 4418\nvalue = [4233, 185]\nclass = no'),
 Text(0.07623626373626374, 0.9318181818181818, 'age <= 0.457\nentropy = 0.07\nsamp
les = 2840\nvalue = [2816, 24]\nclass = no'),
 Text(0.045787545787545784, 0.8863636363636364, 'smoking_status <= 2.5\nentropy =
0.007\nsamples = 1780\nvalue = [1779, 1]\nclass = no'),
 Text(0.040903540903540904, 0.8409090909090909, 'entropy = 0.0\nsamples = 1574\nva
lue = [1574, 0]\nclass = no'),
 Text(0.05067155067155067, 0.8409090909090909, 'avg_glucose_level <= 0.097\nentrop
y = 0.044\nsamples = 206\nvalue = [205, 1]\nclass = no'),
 Text(0.045787545787545784, 0.7954545454545454, 'avg_glucose_level <= 0.096\nentro
py = 0.124\nsamples = 59\nvalue = [58, 1]\nclass = no'),
 Text(0.040903540903540904, 0.75, 'entropy = 0.0\nsamples = 58\nvalue = [58, 0]\nc
lass = no'),
 Text(0.05067155067155067, 0.75, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]\nclas
s = yes'),
 Text(0.05555555555555555, 0.7954545454545454, 'entropy = 0.0\nsamples = 147\nvalu
e = [147, 0]\nclass = no'),
 Text(0.10668498168498168, 0.8863636363636364, 'hypertension <= 0.5\nentropy = 0.1
51\nsamples = 1060\nvalue = [1037, 23]\nclass = no'),
 Text(0.07905982905982906, 0.8409090909090909, 'avg_glucose_level <= 0.231\nentrop
y = 0.133\nsamples = 968\nvalue = [950, 18]\nclass = no'),
 Text(0.06532356532356533, 0.7954545454545454, 'avg_glucose_level <= 0.229\nentrop
y = 0.162\nsamples = 672\nvalue = [656, 16]\nclass = no'),
 Text(0.06043956043956044, 0.75, 'bmi <= 0.242\nentropy = 0.154\nsamples = 671\nva
lue = [656, 15]\nclass = no'),
 Text(0.03785103785103785, 0.7045454545454546, 'bmi <= 0.218\nentropy = 0.204\nsam
ples = 408\nvalue = [395, 13]\nclass = no'),
 Text(0.014652014652014652, 0.6590909090909091, 'avg_glucose_level <= 0.087\nentro
py = 0.133\nsamples = 324\nvalue = [318, 6]\nclass = no'),
 Text(0.009768009768009768, 0.6136363636363636, 'entropy = 0.0\nsamples = 94\nvalu
e = [94, 0]\nclass = no'),
 Text(0.019536019536019536, 0.6136363636363636, 'avg_glucose_level <= 0.088\nentro
py = 0.174\nsamples = 230\nvalue = [224, 6]\nclass = no'),
 Text(0.014652014652014652, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.02442002442002442, 0.5681818181818182, 'bmi <= 0.156\nentropy = 0.152\nsam
ples = 229\nvalue = [224, 5]\nclass = no'),
 Text(0.019536019536019536, 0.5227272727272727, 'entropy = 0.0\nsamples = 69\nvalu
e = [69, 0]\nclass = no'),
 Text(0.029304029304029304, 0.5227272727272727, 'bmi <= 0.195\nentropy = 0.201\nsa
mples = 160\nvalue = [155, 5]\nclass = no'),
 Text(0.02442002442002442, 0.4772727272727273, 'avg_glucose_level <= 0.131\nentrop
y = 0.3\nsamples = 94\nvalue = [89, 5]\nclass = no'),
 Text(0.009768009768009768, 0.4318181818181818, 'avg_glucose_level <= 0.124\nentro
py = 0.575\nsamples = 22\nvalue = [19, 3]\nclass = no'),
 Text(0.004884004884004884, 0.38636363636363635, 'entropy = 0.0\nsamples = 17\nval
ue = [17, 0]\nclass = no'),
 Text(0.014652014652014652, 0.38636363636363635, 'age <= 0.481\nentropy = 0.971\ns
amples = 5\nvalue = [2, 3]\nclass = yes'),
 Text(0.009768009768009768, 0.3409090909090909, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.019536019536019536, 0.3409090909090909, 'smoking_status <= 0.5\nentropy =
0.918\nsamples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.014652014652014652, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalu
e = [0, 1]\nclass = yes'),
 Text(0.02442002442002442, 0.29545454545454547, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.03907203907203907, 0.4318181818181818, 'age <= 0.615\nentropy = 0.183\nsam
ples = 72\nvalue = [70, 2]\nclass = no'),
 Text(0.03418803418803419, 0.38636363636363635, 'entropy = 0.0\nsamples = 58\nvalu
e = [58, 0]\nclass = no'),
 Text(0.04395604395604396, 0.38636363636363635, 'gender <= 0.5\nentropy = 0.592\ns
amples = 14\nvalue = [12, 2]\nclass = no'),
```

```
 Text(0.03907203907203907, 0.3409090909090909, 'bmi <= 0.18\nentropy = 0.971\nsamp
les = 5\nvalue = [3, 2]\nclass = no'),
 Text(0.03418803418803419, 0.29545454545454547, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.04395604395604396, 0.29545454545454547, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.04884004884004884, 0.3409090909090909, 'entropy = 0.0\nsamples = 9\nvalue
= [9, 0]\nclass = no'),
 Text(0.03418803418803419, 0.4772727272727273, 'entropy = 0.0\nsamples = 66\nvalue
= [66, 0]\nclass = no'),
 Text(0.06105006105006105, 0.6590909090909091, 'age <= 0.615\nentropy = 0.414\nsam
ples = 84\nvalue = [77, 7]\nclass = no'),
 Text(0.05616605616605617, 0.6136363636363636, 'age <= 0.542\nentropy = 0.514\nsam
ples = 61\nvalue = [54, 7]\nclass = no'),
 Text(0.04395604395604396, 0.5681818181818182, 'work_type <= 2.5\nentropy = 0.201
\nsamples = 32\nvalue = [31, 1]\nclass = no'),
 Text(0.03907203907203907, 0.5227272727272727, 'entropy = 0.0\nsamples = 30\nvalue
= [30, 0]\nclass = no'),
 Text(0.04884004884004884, 0.5227272727272727, 'avg_glucose_level <= 0.2\nentropy
= 1.0\nsamples = 2\nvalue = [1, 1]\nclass = no'),
 Text(0.04395604395604396, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.05372405372405373, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.06837606837606838, 0.5681818181818182, 'ever_married <= 0.5\nentropy = 0.7
36\nsamples = 29\nvalue = [23, 6]\nclass = no'),
 Text(0.06349206349206349, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.07326007326007326, 0.5227272727272727, 'age <= 0.567\nentropy = 0.677\nsam
ples = 28\nvalue = [23, 5]\nclass = no'),
 Text(0.06349206349206349, 0.4772727272727273, 'bmi <= 0.235\nentropy = 0.954\nsam
ples = 8\nvalue = [5, 3]\nclass = no'),
 Text(0.05860805860805861, 0.4318181818181818, 'gender <= 0.5\nentropy = 1.0\nsamp
les = 6\nvalue = [3, 3]\nclass = no'),
 Text(0.05372405372405373, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.06349206349206349, 0.38636363636363635, 'bmi <= 0.219\nentropy = 0.811\nsa
mples = 4\nvalue = [3, 1]\nclass = no'),
 Text(0.05860805860805861, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.06837606837606838, 0.3409090909090909, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.06837606837606838, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.08302808302808302, 0.4772727272727273, 'age <= 0.591\nentropy = 0.469\nsam
ples = 20\nvalue = [18, 2]\nclass = no'),
 Text(0.07814407814407814, 0.4318181818181818, 'entropy = 0.0\nsamples = 9\nvalue
= [9, 0]\nclass = no'),
 Text(0.08791208791208792, 0.4318181818181818, 'bmi <= 0.239\nentropy = 0.684\nsam
ples = 11\nvalue = [9, 2]\nclass = no'),
 Text(0.08302808302808302, 0.38636363636363635, 'avg_glucose_level <= 0.03\nentrop
y = 0.469\nsamples = 10\nvalue = [9, 1]\nclass = no'),
 Text(0.07814407814407814, 0.3409090909090909, 'avg_glucose_level <= 0.024\nentrop
y = 0.918\nsamples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.07326007326007326, 0.29545454545454547, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.08302808302808302, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.08791208791208792, 0.3409090909090909, 'entropy = 0.0\nsamples = 7\nvalue
= [7, 0]\nclass = no'),
 Text(0.0927960927960928, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.06593406593406594, 0.6136363636363636, 'entropy = 0.0\nsamples = 23\nvalue
= [23, 0]\nclass = no'),
```

```
 Text(0.08302808302808302, 0.7045454545454546, 'bmi <= 0.352\nentropy = 0.064\nsam
ples = 263\nvalue = [261, 2]\nclass = no'),
 Text(0.07814407814407814, 0.6590909090909091, 'entropy = 0.0\nsamples = 196\nvalu
e = [196, 0]\nclass = no'),
 Text(0.08791208791208792, 0.6590909090909091, 'bmi <= 0.358\nentropy = 0.194\nsam
ples = 67\nvalue = [65, 2]\nclass = no'),
 Text(0.08302808302808302, 0.6136363636363636, 'age <= 0.561\nentropy = 0.971\nsam
ples = 5\nvalue = [3, 2]\nclass = no'),
 Text(0.07814407814407814, 0.5681818181818182, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.08791208791208792, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.0927960927960928, 0.6136363636363636, 'entropy = 0.0\nsamples = 62\nvalue
= [62, 0]\nclass = no'),
 Text(0.07020757020757021, 0.75, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]\nclas
s = yes'),
 Text(0.0927960927960928, 0.7954545454545454, 'bmi <= 0.275\nentropy = 0.058\nsamp
les = 296\nvalue = [294, 2]\nclass = no'),
 Text(0.08791208791208792, 0.75, 'entropy = 0.0\nsamples = 203\nvalue = [203, 0]\n
class = no'),
 Text(0.09768009768009768, 0.75, 'bmi <= 0.277\nentropy = 0.15\nsamples = 93\nvalu
e = [91, 2]\nclass = no'),
 Text(0.0927960927960928, 0.7045454545454546, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.10256410256410256, 0.7045454545454546, 'work_type <= 2.5\nentropy = 0.087
\nsamples = 92\nvalue = [91, 1]\nclass = no'),
 Text(0.09768009768009768, 0.6590909090909091, 'entropy = 0.0\nsamples = 77\nvalue
= [77, 0]\nclass = no'),
 Text(0.10744810744810745, 0.6590909090909091, 'smoking_status <= 0.5\nentropy =
0.353\nsamples = 15\nvalue = [14, 1]\nclass = no'),
 Text(0.10256410256410256, 0.6136363636363636, 'gender <= 0.5\nentropy = 0.918\nsa
mples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.09768009768009768, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.10744810744810745, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.11233211233211234, 0.6136363636363636, 'entropy = 0.0\nsamples = 12\nvalue
= [12, 0]\nclass = no'),
 Text(0.1343101343101343, 0.8409090909090909, 'age <= 0.603\nentropy = 0.305\nsamp
les = 92\nvalue = [87, 5]\nclass = no'),
 Text(0.1221001221001221, 0.7954545454545454, 'avg_glucose_level <= 0.018\nentropy
= 0.137\nsamples = 52\nvalue = [51, 1]\nclass = no'),
 Text(0.11721611721611722, 0.75, 'avg_glucose_level <= 0.011\nentropy = 0.918\nsam
ples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.11233211233211234, 0.7045454545454546, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.1221001221001221, 0.7045454545454546, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.12698412698412698, 0.75, 'entropy = 0.0\nsamples = 49\nvalue = [49, 0]\ncl
ass = no'),
 Text(0.14652014652014653, 0.7954545454545454, 'bmi <= 0.365\nentropy = 0.469\nsam
ples = 40\nvalue = [36, 4]\nclass = no'),
 Text(0.13675213675213677, 0.75, 'bmi <= 0.23\nentropy = 0.316\nsamples = 35\nvalu
e = [33, 2]\nclass = no'),
 Text(0.13186813186813187, 0.7045454545454546, 'bmi <= 0.205\nentropy = 0.619\nsam
ples = 13\nvalue = [11, 2]\nclass = no'),
 Text(0.12698412698412698, 0.6590909090909091, 'entropy = 0.0\nsamples = 9\nvalue
= [9, 0]\nclass = no'),
 Text(0.13675213675213677, 0.6590909090909091, 'gender <= 0.5\nentropy = 1.0\nsamp
les = 4\nvalue = [2, 2]\nclass = no'),
 Text(0.13186813186813187, 0.6136363636363636, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.14163614163614163, 0.6136363636363636, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
```

```
 Text(0.14163614163614163, 0.7045454545454546, 'entropy = 0.0\nsamples = 22\nvalue
= [22, 0]\nclass = no'),
 Text(0.1562881562881563, 0.75, 'bmi <= 0.444\nentropy = 0.971\nsamples = 5\nvalue
= [3, 2]\nclass = no'),
 Text(0.1514041514041514, 0.7045454545454546, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.16117216117216118, 0.7045454545454546, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.5144421550671551, 0.9318181818181818, 'age <= 0.823\nentropy = 0.475\nsamp
les = 1578\nvalue = [1417, 161]\nclass = no'),
 Text(0.3046398046398046, 0.8863636363636364, 'avg_glucose_level <= 0.228\nentropy
= 0.32\nsamples = 878\nvalue = [827, 51]\nclass = no'),
 Text(0.2045177045177045, 0.8409090909090909, 'avg_glucose_level <= 0.022\nentropy
= 0.227\nsamples = 546\nvalue = [526, 20]\nclass = no'),
 Text(0.19963369963369965, 0.7954545454545454, 'entropy = 0.0\nsamples = 40\nvalue
= [40, 0]\nclass = no'),
 Text(0.2094017094017094, 0.7954545454545454, 'avg_glucose_level <= 0.022\nentropy
= 0.24\nsamples = 506\nvalue = [486, 20]\nclass = no'),
 Text(0.2045177045177045, 0.75, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]\nclass
= yes'),
 Text(0.21428571428571427, 0.75, 'gender <= 0.5\nentropy = 0.231\nsamples = 505\nv
alue = [486, 19]\nclass = no'),
 Text(0.17094017094017094, 0.7045454545454546, 'bmi <= 0.171\nentropy = 0.17\nsamp
les = 278\nvalue = [271, 7]\nclass = no'),
 Text(0.16605616605616605, 0.6590909090909091, 'entropy = 0.0\nsamples = 65\nvalue
= [65, 0]\nclass = no'),
 Text(0.17582417582417584, 0.6590909090909091, 'bmi <= 0.176\nentropy = 0.209\nsam
ples = 213\nvalue = [206, 7]\nclass = no'),
 Text(0.15506715506715507, 0.6136363636363636, 'work_type <= 2.5\nentropy = 0.811
\nsamples = 8\nvalue = [6, 2]\nclass = no'),
 Text(0.15018315018315018, 0.5681818181818182, 'entropy = 0.0\nsamples = 5\nvalue
= [5, 0]\nclass = no'),
 Text(0.15995115995115994, 0.5681818181818182, 'avg_glucose_level <= 0.18\nentropy
= 0.918\nsamples = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.15506715506715507, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.16483516483516483, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.19658119658119658, 0.6136363636363636, 'ever_married <= 0.5\nentropy = 0.1
65\nsamples = 205\nvalue = [200, 5]\nclass = no'),
 Text(0.1794871794871795, 0.5681818181818182, 'bmi <= 0.309\nentropy = 0.567\nsamp
les = 15\nvalue = [13, 2]\nclass = no'),
 Text(0.1746031746031746, 0.5227272727272727, 'entropy = 0.0\nsamples = 12\nvalue
= [12, 0]\nclass = no'),
 Text(0.18437118437118438, 0.5227272727272727, 'bmi <= 0.353\nentropy = 0.918\nsam
ples = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.1794871794871795, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.18925518925518925, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.21367521367521367, 0.5681818181818182, 'avg_glucose_level <= 0.195\nentrop
y = 0.117\nsamples = 190\nvalue = [187, 3]\nclass = no'),
 Text(0.2039072039072039, 0.5227272727272727, 'hypertension <= 0.5\nentropy = 0.05
5\nsamples = 160\nvalue = [159, 1]\nclass = no'),
 Text(0.199023199023199, 0.4772727272727273, 'entropy = 0.0\nsamples = 145\nvalue
= [145, 0]\nclass = no'),
 Text(0.2087912087912088, 0.4772727272727273, 'avg_glucose_level <= 0.076\nentropy
= 0.353\nsamples = 15\nvalue = [14, 1]\nclass = no'),
 Text(0.2039072039072039, 0.4318181818181818, 'smoking_status <= 2.5\nentropy = 1.
0\nsamples = 2\nvalue = [1, 1]\nclass = no'),
 Text(0.199023199023199, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.2087912087912088, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
```

```
 Text(0.21367521367521367, 0.4318181818181818, 'entropy = 0.0\nsamples = 13\nvalue
= [13, 0]\nclass = no'),
 Text(0.22344322344322345, 0.5227272727272727, 'avg_glucose_level <= 0.195\nentrop
y = 0.353\nsamples = 30\nvalue = [28, 2]\nclass = no'),
 Text(0.21855921855921856, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.22832722832722832, 0.4772727272727273, 'age <= 0.786\nentropy = 0.216\nsam
ples = 29\nvalue = [28, 1]\nclass = no'),
 Text(0.22344322344322345, 0.4318181818181818, 'entropy = 0.0\nsamples = 21\nvalue
= [21, 0]\nclass = no'),
 Text(0.23321123321123322, 0.4318181818181818, 'avg_glucose_level <= 0.213\nentrop
y = 0.544\nsamples = 8\nvalue = [7, 1]\nclass = no'),
 Text(0.22832722832722832, 0.38636363636363635, 'Residence_type <= 0.5\nentropy =
1.0\nsamples = 2\nvalue = [1, 1]\nclass = no'),
 Text(0.22344322344322345, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.23321123321123322, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.23809523809523808, 0.38636363636363635, 'entropy = 0.0\nsamples = 6\nvalue
= [6, 0]\nclass = no'),
 Text(0.2576312576312576, 0.7045454545454546, 'avg_glucose_level <= 0.073\nentropy
= 0.298\nsamples = 227\nvalue = [215, 12]\nclass = no'),
 Text(0.25274725274725274, 0.6590909090909091, 'entropy = 0.0\nsamples = 46\nvalue
= [46, 0]\nclass = no'),
 Text(0.2625152625152625, 0.6590909090909091, 'avg_glucose_level <= 0.075\nentropy
= 0.352\nsamples = 181\nvalue = [169, 12]\nclass = no'),
 Text(0.2576312576312576, 0.6136363636363636, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.2673992673992674, 0.6136363636363636, 'avg_glucose_level <= 0.193\nentropy
= 0.332\nsamples = 180\nvalue = [169, 11]\nclass = no'),
 Text(0.2625152625152625, 0.5681818181818182, 'avg_glucose_level <= 0.193\nentropy
= 0.371\nsamples = 154\nvalue = [143, 11]\nclass = no'),
 Text(0.2576312576312576, 0.5227272727272727, 'smoking_status <= 1.5\nentropy = 0.
348\nsamples = 153\nvalue = [143, 10]\nclass = no'),
 Text(0.24786324786324787, 0.4772727272727273, 'age <= 0.689\nentropy = 0.481\nsam
ples = 77\nvalue = [69, 8]\nclass = no'),
 Text(0.24297924297924298, 0.4318181818181818, 'entropy = 0.0\nsamples = 19\nvalue
= [19, 0]\nclass = no'),
 Text(0.25274725274725274, 0.4318181818181818, 'bmi <= 0.372\nentropy = 0.579\nsam
ples = 58\nvalue = [50, 8]\nclass = no'),
 Text(0.24786324786324787, 0.38636363636363635, 'avg_glucose_level <= 0.174\nentro
py = 0.537\nsamples = 57\nvalue = [50, 7]\nclass = no'),
 Text(0.24297924297924298, 0.3409090909090909, 'avg_glucose_level <= 0.173\nentrop
y = 0.592\nsamples = 49\nvalue = [42, 7]\nclass = no'),
 Text(0.23809523809523808, 0.29545454545454547, 'avg_glucose_level <= 0.144\nentro
py = 0.544\nsamples = 48\nvalue = [42, 6]\nclass = no'),
 Text(0.23321123321123322, 0.25, 'avg_glucose_level <= 0.143\nentropy = 0.661\nsam
ples = 35\nvalue = [29, 6]\nclass = no'),
 Text(0.22832722832722832, 0.20454545454545456, 'work_type <= 1.0\nentropy = 0.533
\nsamples = 33\nvalue = [29, 4]\nclass = no'),
 Text(0.21611721611721613, 0.1590909090909091, 'bmi <= 0.237\nentropy = 0.971\nsam
ples = 5\nvalue = [3, 2]\nclass = no'),
 Text(0.21123321123321123, 0.11363636363636363, 'avg_glucose_level <= 0.121\nentro
py = 0.918\nsamples = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.20634920634920634, 0.06818181818181818, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.21611721611721613, 0.06818181818181818, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.221001221001221, 0.11363636363636363, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.24053724053724054, 0.1590909090909091, 'bmi <= 0.131\nentropy = 0.371\nsam
ples = 28\nvalue = [26, 2]\nclass = no'),
 Text(0.23076923076923078, 0.11363636363636363, 'age <= 0.75\nentropy = 1.0\nsampl
es = 2\nvalue = [1, 1]\nclass = no'),
```

```
 Text(0.2258852258852259, 0.06818181818181818, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.23565323565323565, 0.06818181818181818, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.2503052503052503, 0.11363636363636363, 'avg_glucose_level <= 0.137\nentrop
y = 0.235\nsamples = 26\nvalue = [25, 1]\nclass = no'),
 Text(0.2454212454212454, 0.06818181818181818, 'entropy = 0.0\nsamples = 23\nvalue
= [23, 0]\nclass = no'),
 Text(0.25518925518925517, 0.06818181818181818, 'avg_glucose_level <= 0.138\nentro
py = 0.918\nsamples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.2503052503052503, 0.022727272727272728, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.2600732600732601, 0.022727272727272728, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.23809523809523808, 0.20454545454545456, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.24297924297924298, 0.25, 'entropy = 0.0\nsamples = 13\nvalue = [13, 0]\ncl
ass = no'),
 Text(0.24786324786324787, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.25274725274725274, 0.3409090909090909, 'entropy = 0.0\nsamples = 8\nvalue
= [8, 0]\nclass = no'),
 Text(0.2576312576312576, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.2673992673992674, 0.4772727272727273, 'smoking_status <= 2.5\nentropy = 0.
176\nsamples = 76\nvalue = [74, 2]\nclass = no'),
 Text(0.2625152625152625, 0.4318181818181818, 'entropy = 0.0\nsamples = 49\nvalue
= [49, 0]\nclass = no'),
 Text(0.27228327228327226, 0.4318181818181818, 'Residence_type <= 0.5\nentropy =
0.381\nsamples = 27\nvalue = [25, 2]\nclass = no'),
 Text(0.2673992673992674, 0.38636363636363635, 'bmi <= 0.246\nentropy = 0.65\nsamp
les = 12\nvalue = [10, 2]\nclass = no'),
 Text(0.2625152625152625, 0.3409090909090909, 'age <= 0.664\nentropy = 0.863\nsamp
les = 7\nvalue = [5, 2]\nclass = no'),
 Text(0.2576312576312576, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.2673992673992674, 0.29545454545454547, 'hypertension <= 0.5\nentropy = 0.6
5\nsamples = 6\nvalue = [5, 1]\nclass = no'),
 Text(0.2625152625152625, 0.25, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]\nclass
= no'),
 Text(0.27228327228327226, 0.25, 'work_type <= 2.5\nentropy = 1.0\nsamples = 2\nva
lue = [1, 1]\nclass = no'),
 Text(0.2673992673992674, 0.20454545454545456, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.2771672771672772, 0.20454545454545456, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.27228327228327226, 0.3409090909090909, 'entropy = 0.0\nsamples = 5\nvalue
= [5, 0]\nclass = no'),
 Text(0.2771672771672772, 0.38636363636363635, 'entropy = 0.0\nsamples = 15\nvalue
= [15, 0]\nclass = no'),
 Text(0.2673992673992674, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.27228327228327226, 0.5681818181818182, 'entropy = 0.0\nsamples = 26\nvalue
= [26, 0]\nclass = no'),
 Text(0.40476190476190477, 0.8409090909090909, 'heart_disease <= 0.5\nentropy = 0.
448\nsamples = 332\nvalue = [301, 31]\nclass = no'),
 Text(0.3711843711843712, 0.7954545454545454, 'avg_glucose_level <= 0.228\nentropy
= 0.381\nsamples = 297\nvalue = [275, 22]\nclass = no'),
 Text(0.3663003663003663, 0.75, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]\nclass
= yes'),
 Text(0.37606837606837606, 0.75, 'work_type <= 2.5\nentropy = 0.369\nsamples = 296
\nvalue = [275, 21]\nclass = no'),
 Text(0.34798534798534797, 0.7045454545454546, 'smoking_status <= 1.5\nentropy =
0.42\nsamples = 235\nvalue = [215, 20]\nclass = no'),
```

```
 Text(0.3235653235653236, 0.6590909090909091, 'avg_glucose_level <= 0.702\nentropy
= 0.561\nsamples = 99\nvalue = [86, 13]\nclass = no'),
 Text(0.31868131868131866, 0.6136363636363636, 'avg_glucose_level <= 0.693\nentrop
y = 0.681\nsamples = 72\nvalue = [59, 13]\nclass = no'),
 Text(0.3137973137973138, 0.5681818181818182, 'age <= 0.677\nentropy = 0.627\nsamp
les = 70\nvalue = [59, 11]\nclass = no'),
 Text(0.3089133089133089, 0.5227272727272727, 'entropy = 0.0\nsamples = 12\nvalue
= [12, 0]\nclass = no'),
 Text(0.31868131868131866, 0.5227272727272727, 'age <= 0.725\nentropy = 0.701\nsam
ples = 58\nvalue = [47, 11]\nclass = no'),
 Text(0.30158730158730157, 0.4772727272727273, 'smoking_status <= 0.5\nentropy =
0.918\nsamples = 21\nvalue = [14, 7]\nclass = no'),
 Text(0.29181929181929184, 0.4318181818181818, 'work_type <= 1.0\nentropy = 0.65\n
samples = 12\nvalue = [10, 2]\nclass = no'),
 Text(0.2869352869352869, 0.38636363636363635, 'bmi <= 0.219\nentropy = 0.971\nsam
ples = 5\nvalue = [3, 2]\nclass = no'),
 Text(0.28205128205128205, 0.3409090909090909, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.29181929181929184, 0.3409090909090909, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.2967032967032967, 0.38636363636363635, 'entropy = 0.0\nsamples = 7\nvalue
= [7, 0]\nclass = no'),
 Text(0.31135531135531136, 0.4318181818181818, 'avg_glucose_level <= 0.656\nentrop
y = 0.991\nsamples = 9\nvalue = [4, 5]\nclass = yes'),
 Text(0.3064713064713065, 0.38636363636363635, 'avg_glucose_level <= 0.237\nentrop
y = 0.985\nsamples = 7\nvalue = [4, 3]\nclass = no'),
 Text(0.30158730158730157, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.31135531135531136, 0.3409090909090909, 'avg_glucose_level <= 0.26\nentropy
= 1.0\nsamples = 6\nvalue = [3, 3]\nclass = no'),
 Text(0.3064713064713065, 0.29545454545454547, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.3162393162393162, 0.29545454545454547, 'bmi <= 0.318\nentropy = 0.811\nsam
ples = 4\nvalue = [3, 1]\nclass = no'),
 Text(0.31135531135531136, 0.25, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]\nclas
s = no'),
 Text(0.32112332112332115, 0.25, 'bmi <= 0.363\nentropy = 1.0\nsamples = 2\nvalue
= [1, 1]\nclass = no'),
 Text(0.3162393162393162, 0.20454545454545456, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.326007326007326, 0.20454545454545456, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.3162393162393162, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.33577533577533575, 0.4772727272727273, 'age <= 0.762\nentropy = 0.494\nsam
ples = 37\nvalue = [33, 4]\nclass = no'),
 Text(0.3308913308913309, 0.4318181818181818, 'entropy = 0.0\nsamples = 17\nvalue
= [17, 0]\nclass = no'),
 Text(0.34065934065934067, 0.4318181818181818, 'bmi <= 0.254\nentropy = 0.722\nsam
ples = 20\nvalue = [16, 4]\nclass = no'),
 Text(0.33577533577533575, 0.38636363636363635, 'bmi <= 0.235\nentropy = 0.918\nsa
mples = 12\nvalue = [8, 4]\nclass = no'),
 Text(0.3308913308913309, 0.3409090909090909, 'bmi <= 0.157\nentropy = 0.722\nsamp
les = 10\nvalue = [8, 2]\nclass = no'),
 Text(0.326007326007326, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.33577533577533575, 0.29545454545454547, 'avg_glucose_level <= 0.429\nentro
py = 0.503\nsamples = 9\nvalue = [8, 1]\nclass = no'),
 Text(0.3308913308913309, 0.25, 'entropy = 0.0\nsamples = 6\nvalue = [6, 0]\nclass
= no'),
 Text(0.34065934065934067, 0.25, 'avg_glucose_level <= 0.512\nentropy = 0.918\nsam
ples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.33577533577533575, 0.20454545454545456, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
```

```
 Text(0.34554334554334554, 0.20454545454545456, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.34065934065934067, 0.3409090909090909, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.34554334554334554, 0.38636363636363635, 'entropy = 0.0\nsamples = 8\nvalue
= [8, 0]\nclass = no'),
 Text(0.3235653235653236, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.32844932844932845, 0.6136363636363636, 'entropy = 0.0\nsamples = 27\nvalue
= [27, 0]\nclass = no'),
 Text(0.3724053724053724, 0.6590909090909091, 'avg_glucose_level <= 0.707\nentropy
= 0.293\nsamples = 136\nvalue = [129, 7]\nclass = no'),
 Text(0.3601953601953602, 0.6136363636363636, 'bmi <= 0.2\nentropy = 0.14\nsamples
= 101\nvalue = [99, 2]\nclass = no'),
 Text(0.3553113553113553, 0.5681818181818182, 'bmi <= 0.199\nentropy = 0.371\nsamp
les = 28\nvalue = [26, 2]\nclass = no'),
 Text(0.3504273504273504, 0.5227272727272727, 'smoking_status <= 2.5\nentropy = 0.
229\nsamples = 27\nvalue = [26, 1]\nclass = no'),
 Text(0.34554334554334554, 0.4772727272727273, 'entropy = 0.0\nsamples = 18\nvalue
= [18, 0]\nclass = no'),
 Text(0.3553113553113553, 0.4772727272727273, 'bmi <= 0.153\nentropy = 0.503\nsamp
les = 9\nvalue = [8, 1]\nclass = no'),
 Text(0.3504273504273504, 0.4318181818181818, 'entropy = 0.0\nsamples = 7\nvalue =
[7, 0]\nclass = no'),
 Text(0.3601953601953602, 0.4318181818181818, 'age <= 0.701\nentropy = 1.0\nsample
s = 2\nvalue = [1, 1]\nclass = no'),
 Text(0.3553113553113553, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.36507936507936506, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.3601953601953602, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.36507936507936506, 0.5681818181818182, 'entropy = 0.0\nsamples = 73\nvalue
= [73, 0]\nclass = no'),
 Text(0.38461538461538464, 0.6136363636363636, 'avg_glucose_level <= 0.726\nentrop
y = 0.592\nsamples = 35\nvalue = [30, 5]\nclass = no'),
 Text(0.37484737484737485, 0.5681818181818182, 'age <= 0.774\nentropy = 0.918\nsam
ples = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.36996336996337, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.3797313797313797, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.39438339438339437, 0.5681818181818182, 'bmi <= 0.318\nentropy = 0.449\nsam
ples = 32\nvalue = [29, 3]\nclass = no'),
 Text(0.3894993894993895, 0.5227272727272727, 'work_type <= 1.0\nentropy = 0.559\n
samples = 23\nvalue = [20, 3]\nclass = no'),
 Text(0.38461538461538464, 0.4772727272727273, 'entropy = 0.0\nsamples = 7\nvalue
= [7, 0]\nclass = no'),
 Text(0.39438339438339437, 0.4772727272727273, 'avg_glucose_level <= 0.892\nentrop
y = 0.696\nsamples = 16\nvalue = [13, 3]\nclass = no'),
 Text(0.3894993894993895, 0.4318181818181818, 'bmi <= 0.305\nentropy = 0.567\nsamp
les = 15\nvalue = [13, 2]\nclass = no'),
 Text(0.38461538461538464, 0.38636363636363635, 'age <= 0.762\nentropy = 0.371\nsa
mples = 14\nvalue = [13, 1]\nclass = no'),
 Text(0.3797313797313797, 0.3409090909090909, 'entropy = 0.0\nsamples = 11\nvalue
= [11, 0]\nclass = no'),
 Text(0.3894993894993895, 0.3409090909090909, 'smoking_status <= 2.5\nentropy = 0.
918\nsamples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.38461538461538464, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.39438339438339437, 0.29545454545454547, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.39438339438339437, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
```

```
 Text(0.3992673992673993, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.3992673992673993, 0.5227272727272727, 'entropy = 0.0\nsamples = 9\nvalue =
[9, 0]\nclass = no'),
 Text(0.40415140415140416, 0.7045454545454546, 'age <= 0.664\nentropy = 0.121\nsam
ples = 61\nvalue = [60, 1]\nclass = no'),
 Text(0.3992673992673993, 0.6590909090909091, 'bmi <= 0.343\nentropy = 0.722\nsamp
les = 5\nvalue = [4, 1]\nclass = no'),
 Text(0.39438339438339437, 0.6136363636363636, 'entropy = 0.0\nsamples = 4\nvalue
= [4, 0]\nclass = no'),
 Text(0.40415140415140416, 0.6136363636363636, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.409035409035409, 0.6590909090909091, 'entropy = 0.0\nsamples = 56\nvalue =
[56, 0]\nclass = no'),
 Text(0.43833943833943834, 0.7954545454545454, 'smoking_status <= 2.5\nentropy =
0.822\nsamples = 35\nvalue = [26, 9]\nclass = no'),
 Text(0.42857142857142855, 0.75, 'smoking_status <= 1.5\nentropy = 0.544\nsamples
= 24\nvalue = [21, 3]\nclass = no'),
 Text(0.4236874236874237, 0.7045454545454546, 'Residence_type <= 0.5\nentropy = 0.
696\nsamples = 16\nvalue = [13, 3]\nclass = no'),
 Text(0.4188034188034188, 0.6590909090909091, 'entropy = 0.0\nsamples = 7\nvalue =
[7, 0]\nclass = no'),
 Text(0.42857142857142855, 0.6590909090909091, 'bmi <= 0.304\nentropy = 0.918\nsam
ples = 9\nvalue = [6, 3]\nclass = no'),
 Text(0.4236874236874237, 0.6136363636363636, 'smoking_status <= 0.5\nentropy = 1.
0\nsamples = 6\nvalue = [3, 3]\nclass = no'),
 Text(0.4188034188034188, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.42857142857142855, 0.5681818181818182, 'bmi <= 0.269\nentropy = 0.971\nsam
ples = 5\nvalue = [3, 2]\nclass = no'),
 Text(0.4236874236874237, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.43345543345543347, 0.5227272727272727, 'hypertension <= 0.5\nentropy = 0.9
18\nsamples = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.42857142857142855, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.43833943833943834, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.43345543345543347, 0.6136363636363636, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.43345543345543347, 0.7045454545454546, 'entropy = 0.0\nsamples = 8\nvalue
= [8, 0]\nclass = no'),
 Text(0.4481074481074481, 0.75, 'bmi <= 0.295\nentropy = 0.994\nsamples = 11\nvalu
e = [5, 6]\nclass = yes'),
 Text(0.4432234432234432, 0.7045454545454546, 'gender <= 0.5\nentropy = 0.863\nsam
ples = 7\nvalue = [5, 2]\nclass = no'),
 Text(0.43833943833943834, 0.6590909090909091, 'entropy = 0.0\nsamples = 4\nvalue
= [4, 0]\nclass = no'),
 Text(0.4481074481074481, 0.6590909090909091, 'bmi <= 0.19\nentropy = 0.918\nsampl
es = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.4432234432234432, 0.6136363636363636, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.452991452991453, 0.6136363636363636, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.452991452991453, 0.7045454545454546, 'entropy = 0.0\nsamples = 4\nvalue =
[0, 4]\nclass = yes'),
 Text(0.7242445054945055, 0.8863636363636364, 'avg_glucose_level <= 0.331\nentropy
= 0.627\nsamples = 700\nvalue = [590, 110]\nclass = no'),
 Text(0.6013431013431013, 0.8409090909090909, 'age <= 0.945\nentropy = 0.543\nsamp
les = 473\nvalue = [414, 59]\nclass = no'),
 Text(0.5036630036630036, 0.7954545454545454, 'Residence_type <= 0.5\nentropy = 0.
402\nsamples = 250\nvalue = [230, 20]\nclass = no'),
 Text(0.4822954822954823, 0.75, 'avg_glucose_level <= 0.198\nentropy = 0.237\nsamp
les = 129\nvalue = [124, 5]\nclass = no'),
```

```
 Text(0.47741147741147744, 0.7045454545454546, 'avg_glucose_level <= 0.197\nentrop
y = 0.307\nsamples = 91\nvalue = [86, 5]\nclass = no'),
 Text(0.4725274725274725, 0.6590909090909091, 'smoking_status <= 1.5\nentropy = 0.
262\nsamples = 90\nvalue = [86, 4]\nclass = no'),
 Text(0.46764346764346765, 0.6136363636363636, 'bmi <= 0.258\nentropy = 0.439\nsam
ples = 44\nvalue = [40, 4]\nclass = no'),
 Text(0.4627594627594628, 0.5681818181818182, 'avg_glucose_level <= 0.111\nentropy
= 0.533\nsamples = 33\nvalue = [29, 4]\nclass = no'),
 Text(0.452991452991453, 0.5227272727272727, 'gender <= 0.5\nentropy = 0.779\nsamp
les = 13\nvalue = [10, 3]\nclass = no'),
 Text(0.4481074481074481, 0.4772727272727273, 'age <= 0.847\nentropy = 0.954\nsamp
les = 8\nvalue = [5, 3]\nclass = no'),
 Text(0.4432234432234432, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.452991452991453, 0.4318181818181818, 'bmi <= 0.145\nentropy = 1.0\nsamples
= 6\nvalue = [3, 3]\nclass = no'),
 Text(0.4481074481074481, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue
= [2, 0]\nclass = no'),
 Text(0.45787545787545786, 0.38636363636363635, 'avg_glucose_level <= 0.059\nentro
py = 0.811\nsamples = 4\nvalue = [1, 3]\nclass = yes'),
 Text(0.452991452991453, 0.3409090909090909, 'bmi <= 0.238\nentropy = 1.0\nsamples
= 2\nvalue = [1, 1]\nclass = no'),
 Text(0.4481074481074481, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue
= [1, 0]\nclass = no'),
 Text(0.45787545787545786, 0.29545454545454547, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.4627594627594628, 0.3409090909090909, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.45787545787545786, 0.4772727272727273, 'entropy = 0.0\nsamples = 5\nvalue
= [5, 0]\nclass = no'),
 Text(0.4725274725274725, 0.5227272727272727, 'avg_glucose_level <= 0.178\nentropy
= 0.286\nsamples = 20\nvalue = [19, 1]\nclass = no'),
 Text(0.46764346764346765, 0.4772727272727273, 'entropy = 0.0\nsamples = 16\nvalue
= [16, 0]\nclass = no'),
 Text(0.47741147741147744, 0.4772727272727273, 'bmi <= 0.199\nentropy = 0.811\nsam
ples = 4\nvalue = [3, 1]\nclass = no'),
 Text(0.4725274725274725, 0.4318181818181818, 'entropy = 0.0\nsamples = 3\nvalue =
[3, 0]\nclass = no'),
 Text(0.4822954822954823, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.4725274725274725, 0.5681818181818182, 'entropy = 0.0\nsamples = 11\nvalue
= [11, 0]\nclass = no'),
 Text(0.47741147741147744, 0.6136363636363636, 'entropy = 0.0\nsamples = 46\nvalue
= [46, 0]\nclass = no'),
 Text(0.4822954822954823, 0.6590909090909091, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.48717948717948717, 0.7045454545454546, 'entropy = 0.0\nsamples = 38\nvalue
= [38, 0]\nclass = no'),
 Text(0.525030525030525, 0.75, 'avg_glucose_level <= 0.069\nentropy = 0.541\nsampl
es = 121\nvalue = [106, 15]\nclass = no'),
 Text(0.5201465201465202, 0.7045454545454546, 'entropy = 0.0\nsamples = 22\nvalue
= [22, 0]\nclass = no'),
 Text(0.5299145299145299, 0.7045454545454546, 'avg_glucose_level <= 0.129\nentropy
= 0.614\nsamples = 99\nvalue = [84, 15]\nclass = no'),
 Text(0.5018315018315018, 0.6590909090909091, 'age <= 0.915\nentropy = 0.904\nsamp
les = 25\nvalue = [17, 8]\nclass = no'),
 Text(0.49694749694749696, 0.6136363636363636, 'work_type <= 2.5\nentropy = 0.982
\nsamples = 19\nvalue = [11, 8]\nclass = no'),
 Text(0.48717948717948717, 0.5681818181818182, 'work_type <= 1.0\nentropy = 0.811
\nsamples = 12\nvalue = [9, 3]\nclass = no'),
 Text(0.4822954822954823, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.49206349206349204, 0.5227272727272727, 'avg_glucose_level <= 0.071\nentrop
y = 0.469\nsamples = 10\nvalue = [9, 1]\nclass = no'),
```

```
 Text(0.48717948717948717, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.49694749694749696, 0.4772727272727273, 'entropy = 0.0\nsamples = 9\nvalue
= [9, 0]\nclass = no'),
 Text(0.5067155067155067, 0.5681818181818182, 'smoking_status <= 1.5\nentropy = 0.
863\nsamples = 7\nvalue = [2, 5]\nclass = yes'),
 Text(0.5018315018315018, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.5115995115995116, 0.5227272727272727, 'bmi <= 0.158\nentropy = 0.65\nsampl
es = 6\nvalue = [1, 5]\nclass = yes'),
 Text(0.5067155067155067, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.5164835164835165, 0.4772727272727273, 'entropy = 0.0\nsamples = 5\nvalue =
[0, 5]\nclass = yes'),
 Text(0.5067155067155067, 0.6136363636363636, 'entropy = 0.0\nsamples = 6\nvalue =
[6, 0]\nclass = no'),
 Text(0.557997557997558, 0.6590909090909091, 'age <= 0.921\nentropy = 0.452\nsampl
es = 74\nvalue = [67, 7]\nclass = no'),
 Text(0.5457875457875457, 0.6136363636363636, 'bmi <= 0.194\nentropy = 0.33\nsampl
es = 66\nvalue = [62, 4]\nclass = no'),
 Text(0.536019536019536, 0.5681818181818182, 'work_type <= 2.5\nentropy = 0.592\ns
amples = 21\nvalue = [18, 3]\nclass = no'),
 Text(0.5311355311355311, 0.5227272727272727, 'avg_glucose_level <= 0.177\nentropy
= 0.845\nsamples = 11\nvalue = [8, 3]\nclass = no'),
 Text(0.5262515262515263, 0.4772727272727273, 'entropy = 0.0\nsamples = 4\nvalue =
[4, 0]\nclass = no'),
 Text(0.536019536019536, 0.4772727272727273, 'avg_glucose_level <= 0.201\nentropy
= 0.985\nsamples = 7\nvalue = [4, 3]\nclass = no'),
 Text(0.5311355311355311, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.5409035409035409, 0.4318181818181818, 'bmi <= 0.193\nentropy = 0.722\nsamp
les = 5\nvalue = [4, 1]\nclass = no'),
 Text(0.536019536019536, 0.38636363636363635, 'entropy = 0.0\nsamples = 4\nvalue =
[4, 0]\nclass = no'),
 Text(0.5457875457875457, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.5409035409035409, 0.5227272727272727, 'entropy = 0.0\nsamples = 10\nvalue
= [10, 0]\nclass = no'),
 Text(0.5555555555555556, 0.5681818181818182, 'hypertension <= 0.5\nentropy = 0.15
4\nsamples = 45\nvalue = [44, 1]\nclass = no'),
 Text(0.5506715506715507, 0.5227272727272727, 'entropy = 0.0\nsamples = 35\nvalue
= [35, 0]\nclass = no'),
 Text(0.5604395604395604, 0.5227272727272727, 'avg_glucose_level <= 0.172\nentropy
= 0.469\nsamples = 10\nvalue = [9, 1]\nclass = no'),
 Text(0.5555555555555556, 0.4772727272727273, 'smoking_status <= 1.5\nentropy = 0.
918\nsamples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.5506715506715507, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.5604395604395604, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.5653235653235653, 0.4772727272727273, 'entropy = 0.0\nsamples = 7\nvalue =
[7, 0]\nclass = no'),
 Text(0.5702075702075702, 0.6136363636363636, 'avg_glucose_level <= 0.224\nentropy
= 0.954\nsamples = 8\nvalue = [5, 3]\nclass = no'),
 Text(0.5653235653235653, 0.5681818181818182, 'entropy = 0.0\nsamples = 4\nvalue =
[4, 0]\nclass = no'),
 Text(0.575091575091575, 0.5681818181818182, 'bmi <= 0.244\nentropy = 0.811\nsampl
es = 4\nvalue = [1, 3]\nclass = yes'),
 Text(0.5702075702075702, 0.5227272727272727, 'entropy = 0.0\nsamples = 3\nvalue =
[0, 3]\nclass = yes'),
 Text(0.57997557997558, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.699023199023199, 0.7954545454545454, 'bmi <= 0.186\nentropy = 0.669\nsampl
es = 223\nvalue = [184, 39]\nclass = no'),
```

```
 Text(0.6501831501831502, 0.75, 'avg_glucose_level <= 0.11\nentropy = 0.803\nsampl
es = 94\nvalue = [71, 23]\nclass = no'),
 Text(0.6239316239316239, 0.7045454545454546, 'avg_glucose_level <= 0.08\nentropy
= 0.926\nsamples = 41\nvalue = [27, 14]\nclass = no'),
 Text(0.6043956043956044, 0.6590909090909091, 'bmi <= 0.158\nentropy = 0.784\nsamp
les = 30\nvalue = [23, 7]\nclass = no'),
 Text(0.5897435897435898, 0.6136363636363636, 'heart_disease <= 0.5\nentropy = 0.3
37\nsamples = 16\nvalue = [15, 1]\nclass = no'),
 Text(0.5848595848595849, 0.5681818181818182, 'entropy = 0.0\nsamples = 12\nvalue
= [12, 0]\nclass = no'),
 Text(0.5946275946275946, 0.5681818181818182, 'Residence_type <= 0.5\nentropy = 0.
811\nsamples = 4\nvalue = [3, 1]\nclass = no'),
 Text(0.5897435897435898, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.5995115995115995, 0.5227272727272727, 'entropy = 0.0\nsamples = 3\nvalue =
[3, 0]\nclass = no'),
 Text(0.6190476190476191, 0.6136363636363636, 'hypertension <= 0.5\nentropy = 0.98
5\nsamples = 14\nvalue = [8, 6]\nclass = no'),
 Text(0.6141636141636142, 0.5681818181818182, 'work_type <= 2.5\nentropy = 0.918\n
samples = 12\nvalue = [8, 4]\nclass = no'),
 Text(0.6092796092796092, 0.5227272727272727, 'avg_glucose_level <= 0.018\nentropy
= 1.0\nsamples = 8\nvalue = [4, 4]\nclass = no'),
 Text(0.6043956043956044, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.6141636141636142, 0.4772727272727273, 'avg_glucose_level <= 0.043\nentropy
= 0.985\nsamples = 7\nvalue = [4, 3]\nclass = no'),
 Text(0.6092796092796092, 0.4318181818181818, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.6190476190476191, 0.4318181818181818, 'age <= 0.969\nentropy = 0.971\nsamp
les = 5\nvalue = [2, 3]\nclass = yes'),
 Text(0.6141636141636142, 0.38636363636363635, 'avg_glucose_level <= 0.059\nentrop
y = 0.918\nsamples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.6092796092796092, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.6190476190476191, 0.3409090909090909, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.6239316239316239, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.6190476190476191, 0.5227272727272727, 'entropy = 0.0\nsamples = 4\nvalue =
[4, 0]\nclass = no'),
 Text(0.6239316239316239, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.6434676434676435, 0.6590909090909091, 'age <= 0.982\nentropy = 0.946\nsamp
les = 11\nvalue = [4, 7]\nclass = yes'),
 Text(0.6385836385836385, 0.6136363636363636, 'work_type <= 2.5\nentropy = 1.0\nsa
mples = 8\nvalue = [4, 4]\nclass = no'),
 Text(0.6336996336996337, 0.5681818181818182, 'bmi <= 0.156\nentropy = 0.918\nsamp
les = 6\nvalue = [2, 4]\nclass = yes'),
 Text(0.6288156288156288, 0.5227272727272727, 'entropy = 0.0\nsamples = 3\nvalue =
[0, 3]\nclass = yes'),
 Text(0.6385836385836385, 0.5227272727272727, 'avg_glucose_level <= 0.09\nentropy
= 0.918\nsamples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.6336996336996337, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.6434676434676435, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.6434676434676435, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.6483516483516484, 0.6136363636363636, 'entropy = 0.0\nsamples = 3\nvalue =
[0, 3]\nclass = yes'),
 Text(0.6764346764346765, 0.7045454545454546, 'ever_married <= 0.5\nentropy = 0.65
7\nsamples = 53\nvalue = [44, 9]\nclass = no'),
 Text(0.663003663003663, 0.6590909090909091, 'work_type <= 2.5\nentropy = 0.991\ns
amples = 9\nvalue = [5, 4]\nclass = no'),
```

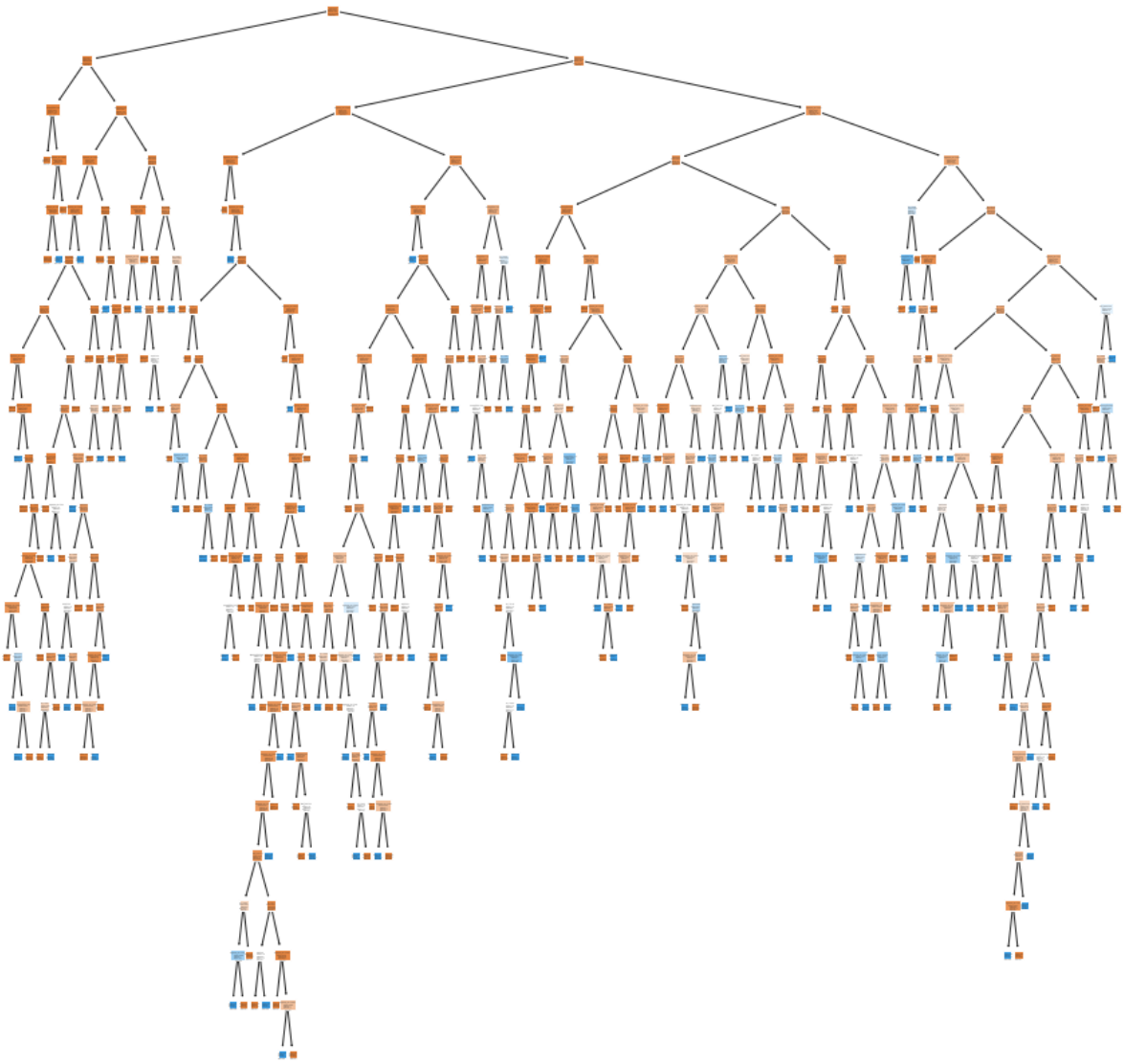
```
 Text(0.6581196581196581, 0.6136363636363636, 'bmi <= 0.134\nentropy = 0.918\nsamp
les = 6\nvalue = [2, 4]\nclass = yes'),
 Text(0.6532356532356532, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.663003663003663, 0.5681818181818182, 'entropy = 0.0\nsamples = 4\nvalue =
[0, 4]\nclass = yes'),
 Text(0.6678876678876678, 0.6136363636363636, 'entropy = 0.0\nsamples = 3\nvalue =
[3, 0]\nclass = no'),
 Text(0.6898656898656899, 0.6590909090909091, 'avg_glucose_level <= 0.187\nentropy
= 0.511\nsamples = 44\nvalue = [39, 5]\nclass = no'),
 Text(0.6776556776556777, 0.6136363636363636, 'bmi <= 0.078\nentropy = 0.211\nsamp
les = 30\nvalue = [29, 1]\nclass = no'),
 Text(0.6727716727716728, 0.5681818181818182, 'bmi <= 0.044\nentropy = 1.0\nsample
s = 2\nvalue = [1, 1]\nclass = no'),
 Text(0.6678876678876678, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.6776556776556777, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.6825396825396826, 0.5681818181818182, 'entropy = 0.0\nsamples = 28\nvalue
= [28, 0]\nclass = no'),
 Text(0.702075702075702, 0.6136363636363636, 'work_type <= 2.5\nentropy = 0.863\ns
amples = 14\nvalue = [10, 4]\nclass = no'),
 Text(0.6923076923076923, 0.5681818181818182, 'bmi <= 0.163\nentropy = 0.971\nsamp
les = 5\nvalue = [2, 3]\nclass = yes'),
 Text(0.6874236874236874, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.6971916971916972, 0.5227272727272727, 'gender <= 0.5\nentropy = 0.918\nsam
ples = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.6923076923076923, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.702075702075702, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.7118437118437119, 0.5681818181818182, 'avg_glucose_level <= 0.19\nentropy
= 0.503\nsamples = 9\nvalue = [8, 1]\nclass = no'),
 Text(0.706959706959707, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.7167277167277167, 0.5227272727272727, 'entropy = 0.0\nsamples = 8\nvalue =
[8, 0]\nclass = no'),
 Text(0.7478632478632479, 0.75, 'smoking_status <= 0.5\nentropy = 0.541\nsamples =
129\nvalue = [113, 16]\nclass = no'),
 Text(0.742979242979243, 0.7045454545454546, 'entropy = 0.0\nsamples = 27\nvalue =
[27, 0]\nclass = no'),
 Text(0.7527472527472527, 0.7045454545454546, 'bmi <= 0.212\nentropy = 0.627\nsamp
les = 102\nvalue = [86, 16]\nclass = no'),
 Text(0.7313797313797313, 0.6590909090909091, 'gender <= 0.5\nentropy = 0.365\nsam
ples = 43\nvalue = [40, 3]\nclass = no'),
 Text(0.7264957264957265, 0.6136363636363636, 'entropy = 0.0\nsamples = 23\nvalue
= [23, 0]\nclass = no'),
 Text(0.7362637362637363, 0.6136363636363636, 'age <= 0.969\nentropy = 0.61\nsampl
es = 20\nvalue = [17, 3]\nclass = no'),
 Text(0.7313797313797313, 0.5681818181818182, 'smoking_status <= 1.5\nentropy = 0.
881\nsamples = 10\nvalue = [7, 3]\nclass = no'),
 Text(0.7264957264957265, 0.5227272727272727, 'entropy = 0.0\nsamples = 4\nvalue =
[4, 0]\nclass = no'),
 Text(0.7362637362637363, 0.5227272727272727, 'hypertension <= 0.5\nentropy = 1.0
\nsamples = 6\nvalue = [3, 3]\nclass = no'),
 Text(0.7313797313797313, 0.4772727272727273, 'avg_glucose_level <= 0.043\nentropy
= 0.811\nsamples = 4\nvalue = [1, 3]\nclass = yes'),
 Text(0.7264957264957265, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.7362637362637363, 0.4318181818181818, 'entropy = 0.0\nsamples = 3\nvalue =
[0, 3]\nclass = yes'),
 Text(0.7411477411477412, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
```

```
 Text(0.7411477411477412, 0.5681818181818182, 'entropy = 0.0\nsamples = 10\nvalue
= [10, 0]\nclass = no'),
 Text(0.7741147741147741, 0.6590909090909091, 'age <= 0.969\nentropy = 0.761\nsamp
les = 59\nvalue = [46, 13]\nclass = no'),
 Text(0.7557997557997558, 0.6136363636363636, 'avg_glucose_level <= 0.238\nentropy
= 0.402\nsamples = 25\nvalue = [23, 2]\nclass = no'),
 Text(0.7509157509157509, 0.5681818181818182, 'entropy = 0.0\nsamples = 21\nvalue
= [21, 0]\nclass = no'),
 Text(0.7606837606837606, 0.5681818181818182, 'avg_glucose_level <= 0.268\nentropy
= 1.0\nsamples = 4\nvalue = [2, 2]\nclass = no'),
 Text(0.7557997557997558, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.7655677655677655, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.7924297924297924, 0.6136363636363636, 'avg_glucose_level <= 0.257\nentropy
= 0.908\nsamples = 34\nvalue = [23, 11]\nclass = no'),
 Text(0.7875457875457875, 0.5681818181818182, 'gender <= 0.5\nentropy = 0.967\nsam
ples = 28\nvalue = [17, 11]\nclass = no'),
 Text(0.7753357753357754, 0.5227272727272727, 'age <= 0.982\nentropy = 0.902\nsamp
les = 22\nvalue = [15, 7]\nclass = no'),
 Text(0.7655677655677655, 0.4772727272727273, 'hypertension <= 0.5\nentropy = 0.99
1\nsamples = 9\nvalue = [4, 5]\nclass = yes'),
 Text(0.7606837606837606, 0.4318181818181818, 'bmi <= 0.27\nentropy = 0.918\nsampl
es = 6\nvalue = [4, 2]\nclass = no'),
 Text(0.7557997557997558, 0.38636363636363635, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.7655677655677655, 0.38636363636363635, 'avg_glucose_level <= 0.042\nentrop
y = 0.918\nsamples = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.7606837606837606, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.7704517704517705, 0.3409090909090909, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.7704517704517705, 0.4318181818181818, 'entropy = 0.0\nsamples = 3\nvalue =
[0, 3]\nclass = yes'),
 Text(0.7851037851037851, 0.4772727272727273, 'Residence_type <= 0.5\nentropy = 0.
619\nsamples = 13\nvalue = [11, 2]\nclass = no'),
 Text(0.7802197802197802, 0.4318181818181818, 'smoking_status <= 1.5\nentropy = 0.
918\nsamples = 6\nvalue = [4, 2]\nclass = no'),
 Text(0.7753357753357754, 0.38636363636363635, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.7851037851037851, 0.38636363636363635, 'ever_married <= 0.5\nentropy = 0.9
18\nsamples = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.7802197802197802, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.78998778998779, 0.3409090909090909, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.78998778998779, 0.4318181818181818, 'entropy = 0.0\nsamples = 7\nvalue =
[7, 0]\nclass = no'),
 Text(0.7997557997557998, 0.5227272727272727, 'avg_glucose_level <= 0.14\nentropy
= 0.918\nsamples = 6\nvalue = [2, 4]\nclass = yes'),
 Text(0.7948717948717948, 0.4772727272727273, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.8046398046398047, 0.4772727272727273, 'entropy = 0.0\nsamples = 4\nvalue =
[0, 4]\nclass = yes'),
 Text(0.7973137973137974, 0.5681818181818182, 'entropy = 0.0\nsamples = 6\nvalue =
[6, 0]\nclass = no'),
 Text(0.8471459096459096, 0.8409090909090909, 'avg_glucose_level <= 0.353\nentropy
= 0.769\nsamples = 227\nvalue = [176, 51]\nclass = no'),
 Text(0.811965811965812, 0.7954545454545454, 'bmi <= 0.222\nentropy = 0.985\nsampl
es = 7\nvalue = [3, 4]\nclass = yes'),
 Text(0.8070818070818071, 0.75, 'smoking_status <= 0.5\nentropy = 0.722\nsamples =
5\nvalue = [1, 4]\nclass = yes'),
 Text(0.8021978021978022, 0.7045454545454546, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
```

```
 Text(0.811965811965812, 0.7045454545454546, 'entropy = 0.0\nsamples = 4\nvalue =
[0, 4]\nclass = yes'),
 Text(0.8168498168498168, 0.75, 'entropy = 0.0\nsamples = 2\nvalue = [2, 0]\nclass
= no'),
 Text(0.8823260073260073, 0.7954545454545454, 'bmi <= 0.177\nentropy = 0.748\nsamp
les = 220\nvalue = [173, 47]\nclass = no'),
 Text(0.8266178266178266, 0.75, 'avg_glucose_level <= 0.659\nentropy = 0.454\nsamp
les = 42\nvalue = [38, 4]\nclass = no'),
 Text(0.8217338217338217, 0.7045454545454546, 'bmi <= 0.158\nentropy = 0.787\nsamp
les = 17\nvalue = [13, 4]\nclass = no'),
 Text(0.8168498168498168, 0.6590909090909091, 'bmi <= 0.152\nentropy = 0.918\nsamp
les = 12\nvalue = [8, 4]\nclass = no'),
 Text(0.811965811965812, 0.6136363636363636, 'avg_glucose_level <= 0.616\nentropy
= 0.722\nsamples = 10\nvalue = [8, 2]\nclass = no'),
 Text(0.8070818070818071, 0.5681818181818182, 'entropy = 0.0\nsamples = 7\nvalue =
[7, 0]\nclass = no'),
 Text(0.8168498168498168, 0.5681818181818182, 'bmi <= 0.142\nentropy = 0.918\nsamp
les = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.811965811965812, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.8217338217338217, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.8217338217338217, 0.6136363636363636, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.8266178266178266, 0.6590909090909091, 'entropy = 0.0\nsamples = 5\nvalue =
[5, 0]\nclass = no'),
 Text(0.8315018315018315, 0.7045454545454546, 'entropy = 0.0\nsamples = 25\nvalue
= [25, 0]\nclass = no'),
 Text(0.938034188034188, 0.75, 'avg_glucose_level <= 0.87\nentropy = 0.798\nsample
s = 178\nvalue = [135, 43]\nclass = no'),
 Text(0.8907203907203908, 0.7045454545454546, 'bmi <= 0.207\nentropy = 0.757\nsamp
les = 165\nvalue = [129, 36]\nclass = no'),
 Text(0.8412698412698413, 0.6590909090909091, 'avg_glucose_level <= 0.584\nentropy
= 0.934\nsamples = 40\nvalue = [26, 14]\nclass = no'),
 Text(0.8315018315018315, 0.6136363636363636, 'age <= 0.994\nentropy = 0.503\nsamp
les = 9\nvalue = [8, 1]\nclass = no'),
 Text(0.8266178266178266, 0.5681818181818182, 'entropy = 0.0\nsamples = 8\nvalue =
[8, 0]\nclass = no'),
 Text(0.8363858363858364, 0.5681818181818182, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.851037851037851, 0.6136363636363636, 'avg_glucose_level <= 0.651\nentropy
= 0.981\nsamples = 31\nvalue = [18, 13]\nclass = no'),
 Text(0.8461538461538461, 0.5681818181818182, 'entropy = 0.0\nsamples = 3\nvalue =
[0, 3]\nclass = yes'),
 Text(0.8559218559218559, 0.5681818181818182, 'avg_glucose_level <= 0.802\nentropy
= 0.94\nsamples = 28\nvalue = [18, 10]\nclass = no'),
 Text(0.8388278388278388, 0.5227272727272727, 'age <= 0.908\nentropy = 0.998\nsamp
les = 19\nvalue = [10, 9]\nclass = no'),
 Text(0.8290598290598291, 0.4772727272727273, 'work_type <= 2.5\nentropy = 0.592\n
samples = 7\nvalue = [6, 1]\nclass = no'),
 Text(0.8241758241758241, 0.4318181818181818, 'entropy = 0.0\nsamples = 6\nvalue =
[6, 0]\nclass = no'),
 Text(0.833943833943834, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.8485958485958486, 0.4772727272727273, 'avg_glucose_level <= 0.718\nentropy
= 0.918\nsamples = 12\nvalue = [4, 8]\nclass = yes'),
 Text(0.8437118437118437, 0.4318181818181818, 'avg_glucose_level <= 0.683\nentropy
= 0.918\nsamples = 6\nvalue = [4, 2]\nclass = no'),
 Text(0.8388278388278388, 0.38636363636363635, 'avg_glucose_level <= 0.66\nentropy
= 0.918\nsamples = 3\nvalue = [1, 2]\nclass = yes'),
 Text(0.833943833943834, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.8437118437118437, 0.3409090909090909, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
```

```
 Text(0.8485958485958486, 0.38636363636363635, 'entropy = 0.0\nsamples = 3\nvalue
= [3, 0]\nclass = no'),
 Text(0.8534798534798534, 0.4318181818181818, 'entropy = 0.0\nsamples = 6\nvalue =
[0, 6]\nclass = yes'),
 Text(0.873015873015873, 0.5227272727272727, 'age <= 0.86\nentropy = 0.503\nsample
s = 9\nvalue = [8, 1]\nclass = no'),
 Text(0.8681318681318682, 0.4772727272727273, 'Residence_type <= 0.5\nentropy = 1.
0\nsamples = 2\nvalue = [1, 1]\nclass = no'),
 Text(0.8632478632478633, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.873015873015873, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue =
[1, 0]\nclass = no'),
 Text(0.8778998778998779, 0.4772727272727273, 'entropy = 0.0\nsamples = 7\nvalue =
[7, 0]\nclass = no'),
 Text(0.9401709401709402, 0.6590909090909091, 'work_type <= 2.5\nentropy = 0.671\n
samples = 125\nvalue = [103, 22]\nclass = no'),
 Text(0.9145299145299145, 0.6136363636363636, 'age <= 0.896\nentropy = 0.783\nsamp
les = 73\nvalue = [56, 17]\nclass = no'),
 Text(0.8876678876678876, 0.5681818181818182, 'smoking_status <= 1.5\nentropy = 0.
533\nsamples = 33\nvalue = [29, 4]\nclass = no'),
 Text(0.8827838827838828, 0.5227272727272727, 'entropy = 0.0\nsamples = 12\nvalue
= [12, 0]\nclass = no'),
 Text(0.8925518925518926, 0.5227272727272727, 'bmi <= 0.405\nentropy = 0.702\nsamp
les = 21\nvalue = [17, 4]\nclass = no'),
 Text(0.8876678876678876, 0.4772727272727273, 'hypertension <= 0.5\nentropy = 0.61
\nsamples = 20\nvalue = [17, 3]\nclass = no'),
 Text(0.8827838827838828, 0.4318181818181818, 'entropy = 0.0\nsamples = 10\nvalue
= [10, 0]\nclass = no'),
 Text(0.8925518925518926, 0.4318181818181818, 'ever_married <= 0.5\nentropy = 0.88
1\nsamples = 10\nvalue = [7, 3]\nclass = no'),
 Text(0.8876678876678876, 0.38636363636363635, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.8974358974358975, 0.38636363636363635, 'age <= 0.872\nentropy = 0.544\nsam
ples = 8\nvalue = [7, 1]\nclass = no'),
 Text(0.8925518925518926, 0.3409090909090909, 'entropy = 0.0\nsamples = 7\nvalue =
[7, 0]\nclass = no'),
 Text(0.9023199023199023, 0.3409090909090909, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.8974358974358975, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.9413919413919414, 0.5681818181818182, 'avg_glucose_level <= 0.805\nentropy
= 0.91\nsamples = 40\nvalue = [27, 13]\nclass = no'),
 Text(0.9365079365079365, 0.5227272727272727, 'bmi <= 0.394\nentropy = 0.842\nsamp
les = 37\nvalue = [27, 10]\nclass = no'),
 Text(0.9316239316239316, 0.4772727272727273, 'bmi <= 0.285\nentropy = 0.811\nsamp
les = 36\nvalue = [27, 9]\nclass = no'),
 Text(0.9267399267399268, 0.4318181818181818, 'bmi <= 0.281\nentropy = 0.894\nsamp
les = 29\nvalue = [20, 9]\nclass = no'),
 Text(0.9218559218559218, 0.38636363636363635, 'bmi <= 0.247\nentropy = 0.863\nsam
ples = 28\nvalue = [20, 8]\nclass = no'),
 Text(0.9120879120879121, 0.3409090909090909, 'bmi <= 0.24\nentropy = 0.964\nsampl
es = 18\nvalue = [11, 7]\nclass = no'),
 Text(0.9072039072039072, 0.29545454545454547, 'Residence_type <= 0.5\nentropy =
0.896\nsamples = 16\nvalue = [11, 5]\nclass = no'),
 Text(0.9023199023199023, 0.25, 'entropy = 0.0\nsamples = 4\nvalue = [4, 0]\nclass
= no'),
 Text(0.9120879120879121, 0.25, 'heart_disease <= 0.5\nentropy = 0.98\nsamples = 1
2\nvalue = [7, 5]\nclass = no'),
 Text(0.9072039072039072, 0.20454545454545456, 'age <= 0.982\nentropy = 0.881\nsam
ples = 10\nvalue = [7, 3]\nclass = no'),
 Text(0.9023199023199023, 0.1590909090909091, 'avg_glucose_level <= 0.409\nentropy
= 0.544\nsamples = 8\nvalue = [7, 1]\nclass = no'),
 Text(0.8974358974358975, 0.11363636363636363, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
```

```
 Text(0.9072039072039072, 0.11363636363636363, 'entropy = 0.0\nsamples = 7\nvalue
= [7, 0]\nclass = no'),
 Text(0.9120879120879121, 0.1590909090909091, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.9169719169719169, 0.20454545454545456, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.9169719169719169, 0.29545454545454547, 'entropy = 0.0\nsamples = 2\nvalue
= [0, 2]\nclass = yes'),
 Text(0.9316239316239316, 0.3409090909090909, 'age <= 0.908\nentropy = 0.469\nsamp
les = 10\nvalue = [9, 1]\nclass = no'),
 Text(0.9267399267399268, 0.29545454545454547, 'Residence_type <= 0.5\nentropy =
1.0\nsamples = 2\nvalue = [1, 1]\nclass = no'),
 Text(0.9218559218559218, 0.25, 'entropy = 0.0\nsamples = 1\nvalue = [0, 1]\nclass
= yes'),
 Text(0.9316239316239316, 0.25, 'entropy = 0.0\nsamples = 1\nvalue = [1, 0]\nclass
= no'),
 Text(0.9365079365079365, 0.29545454545454547, 'entropy = 0.0\nsamples = 8\nvalue
= [8, 0]\nclass = no'),
 Text(0.9316239316239316, 0.38636363636363635, 'entropy = 0.0\nsamples = 1\nvalue
= [0, 1]\nclass = yes'),
 Text(0.9365079365079365, 0.4318181818181818, 'entropy = 0.0\nsamples = 7\nvalue =
[7, 0]\nclass = no'),
 Text(0.9413919413919414, 0.4772727272727273, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.9462759462759462, 0.5227272727272727, 'entropy = 0.0\nsamples = 3\nvalue =
[0, 3]\nclass = yes'),
 Text(0.9658119658119658, 0.6136363636363636, 'avg_glucose_level <= 0.655\nentropy
= 0.457\nsamples = 52\nvalue = [47, 5]\nclass = no'),
 Text(0.960927960927961, 0.5681818181818182, 'bmi <= 0.253\nentropy = 0.896\nsampl
es = 16\nvalue = [11, 5]\nclass = no'),
 Text(0.9560439560439561, 0.5227272727272727, 'entropy = 0.0\nsamples = 6\nvalue =
[6, 0]\nclass = no'),
 Text(0.9658119658119658, 0.5227272727272727, 'gender <= 0.5\nentropy = 1.0\nsampl
es = 10\nvalue = [5, 5]\nclass = no'),
 Text(0.960927960927961, 0.4772727272727273, 'age <= 0.872\nentropy = 0.65\nsample
s = 6\nvalue = [5, 1]\nclass = no'),
 Text(0.9560439560439561, 0.4318181818181818, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.9658119658119658, 0.4318181818181818, 'entropy = 0.0\nsamples = 5\nvalue =
[5, 0]\nclass = no'),
 Text(0.9706959706959707, 0.4772727272727273, 'entropy = 0.0\nsamples = 4\nvalue =
[0, 4]\nclass = yes'),
 Text(0.9706959706959707, 0.5681818181818182, 'entropy = 0.0\nsamples = 36\nvalue
= [36, 0]\nclass = no'),
 Text(0.9853479853479854, 0.7045454545454546, 'heart_disease <= 0.5\nentropy = 0.9
96\nsamples = 13\nvalue = [6, 7]\nclass = yes'),
 Text(0.9804639804639804, 0.6590909090909091, 'bmi <= 0.238\nentropy = 0.918\nsamp
les = 9\nvalue = [6, 3]\nclass = no'),
 Text(0.9755799755799756, 0.6136363636363636, 'entropy = 0.0\nsamples = 4\nvalue =
[4, 0]\nclass = no'),
 Text(0.9853479853479854, 0.6136363636363636, 'Residence_type <= 0.5\nentropy = 0.
971\nsamples = 5\nvalue = [2, 3]\nclass = yes'),
 Text(0.9804639804639804, 0.5681818181818182, 'entropy = 0.0\nsamples = 2\nvalue =
[0, 2]\nclass = yes'),
 Text(0.9902319902319903, 0.5681818181818182, 'age <= 0.902\nentropy = 0.918\nsamp
les = 3\nvalue = [2, 1]\nclass = no'),
 Text(0.9853479853479854, 0.5227272727272727, 'entropy = 0.0\nsamples = 1\nvalue =
[0, 1]\nclass = yes'),
 Text(0.9951159951159951, 0.5227272727272727, 'entropy = 0.0\nsamples = 2\nvalue =
[2, 0]\nclass = no'),
 Text(0.9902319902319903, 0.6590909090909091, 'entropy = 0.0\nsamples = 4\nvalue =
[0, 4]\nclass = yes')]
```



In [40]:
```
print("Accuracy on validation set for Decision Tree model: {:.2%}:".format(metrics
```

```
Accuracy on validation set for Decision Tree model: 91.45%:
```

In [41]:
```
from sklearn.datasets import make_circles
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
```

In [42]:
```
#Hyper-Parameter Tuning
from scipy.stats import randint
from sklearn.model_selection import RandomizedSearchCV
param_dist = {"max_depth": [3, None],
 "min_samples_split": randint(1,9),
 "min_samples_leaf": randint(1, 9),
 }
tree_1 = RandomizedSearchCV(clf, param_dist)
tree_1.fit(X1_train,Y1_train)
```

Out[42]:
```
RandomizedSearchCV(estimator=DecisionTreeClassifier(criterion='entropy'),
                   param_distributions={'max_depth': [3, None],
                                        'min_samples_leaf': <scipy.stats._distn_in
frastructure.rv_frozen object at 0x00000239C15F50A0>,
                                        'min_samples_split': <scipy.stats._distn_i
nfrastructure.rv_frozen object at 0x00000239C10FBD60>})
```

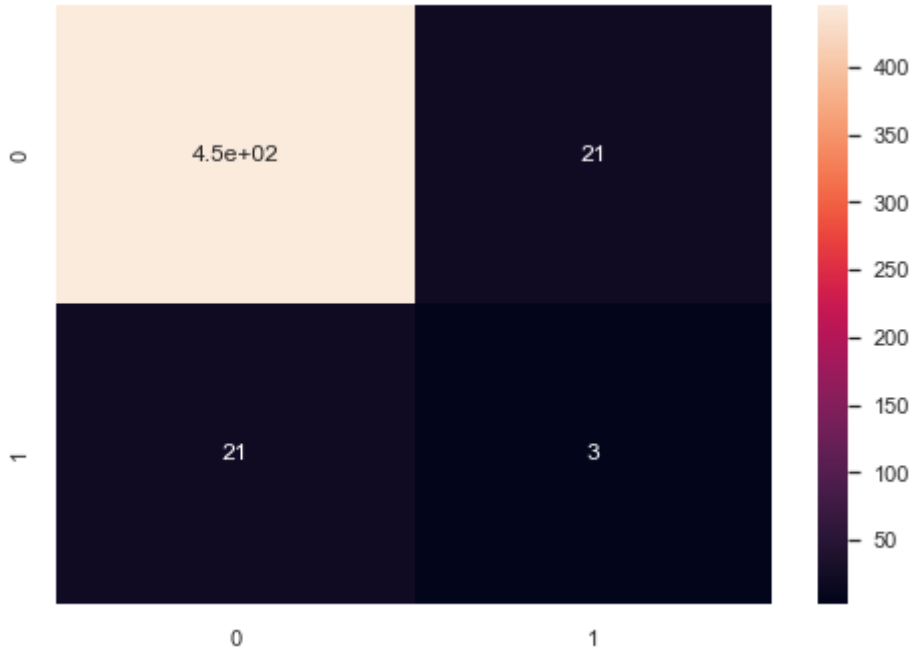
In [43]:
```
print("Tuned Decision Tree Parameters: {}".format(tree_1.best_params_))
print("Best score is {:.2%}:".format(tree_1.best_score_))
```

```
Tuned Decision Tree Parameters: {'max_depth': 3, 'min_samples_leaf': 7, 'min_sampl
es_split': 6}
Best score is 95.81%:
```

In [44]:
```
#Confusion Matrix
from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

print(classification_report(Y1_Val, Y1_pred))

sns.heatmap(confusion_matrix(Y1_Val, Y1_pred),annot=True)
```

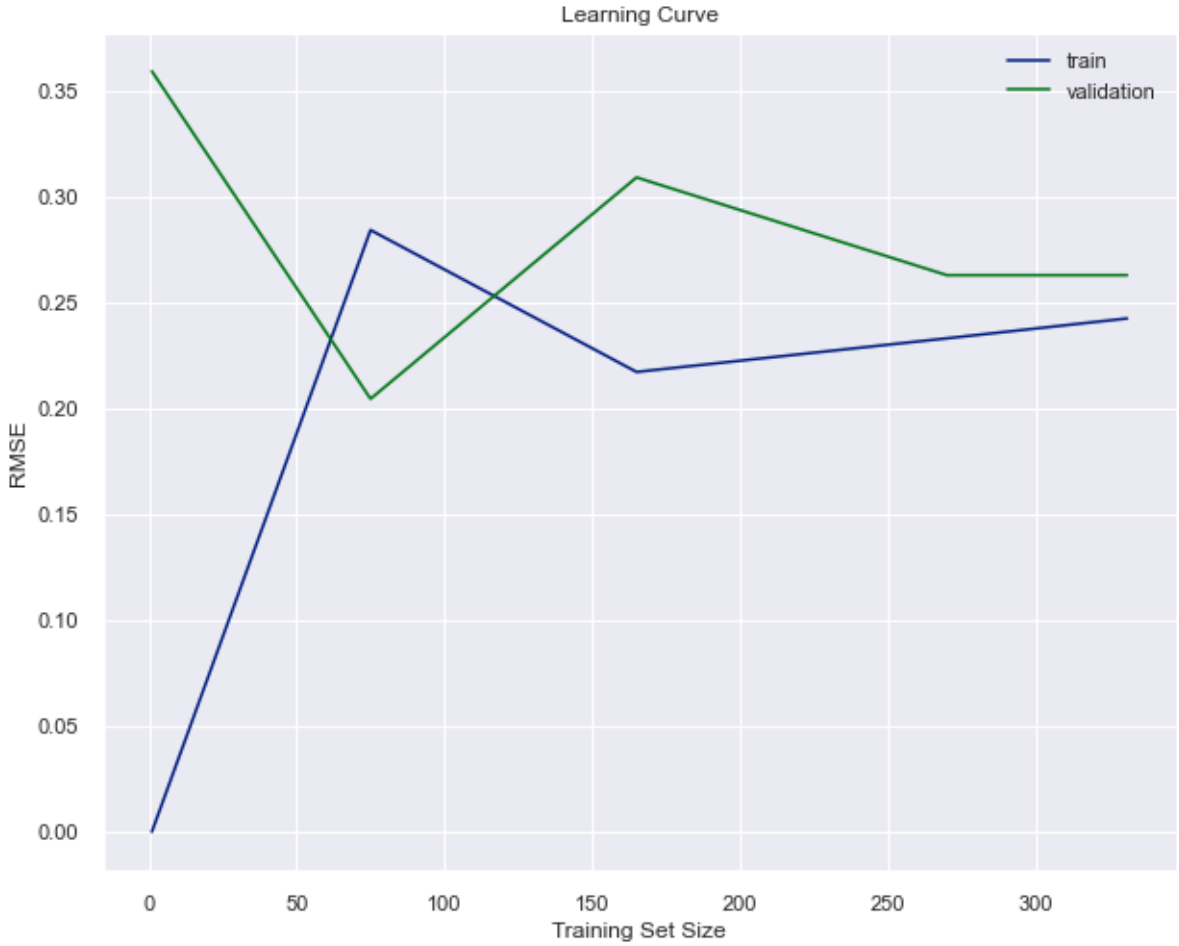
```
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       467
           1       0.12      0.12      0.12        24

    accuracy                           0.91       491
   macro avg       0.54      0.54      0.54       491
weighted avg       0.91      0.91      0.91       491
```

Out[44]:
```
<AxesSubplot:>
```



In [45]:
```
#Learning Curve
from sklearn.model_selection import learning_curve

clf_2_HP = tree.DecisionTreeClassifier(max_depth=3, min_samples_leaf= 8, min_sampl
train_sizes, train_scores, test_scores = learning_curve(
 estimator=clf_2_HP,
 X=X1_train,
 y=Y1_train,
 cv=5,
 scoring="neg_root_mean_squared_error",
 train_sizes = [1, 75, 165, 270, 331]
)
train_mean = -train_scores.mean(axis=1)
test_mean = -test_scores.mean(axis=1)
plt.subplots(figsize=(10,8))
```

```
plt.plot(train_sizes, train_mean, label="train")
plt.plot(train_sizes, test_mean, label="validation")
plt.title("Learning Curve")
plt.xlabel("Training Set Size")
plt.ylabel("RMSE")
plt.legend(loc="best")
plt.show()
```



XG Boost

In [46]:
```
from xgboost import XGBClassifier

xgb_clf = XGBClassifier()
xgb_clf.fit(X1_train,Y1_train)
Y1_pred_xg = xgb_clf.predict(X1_Val)
```

In [47]:
```
score_xgb = xgb_clf.score(X1_Val, Y1_Val)
print("Accuracy on validation set for XGB model: {:.2%}".format(score_xgb))
```

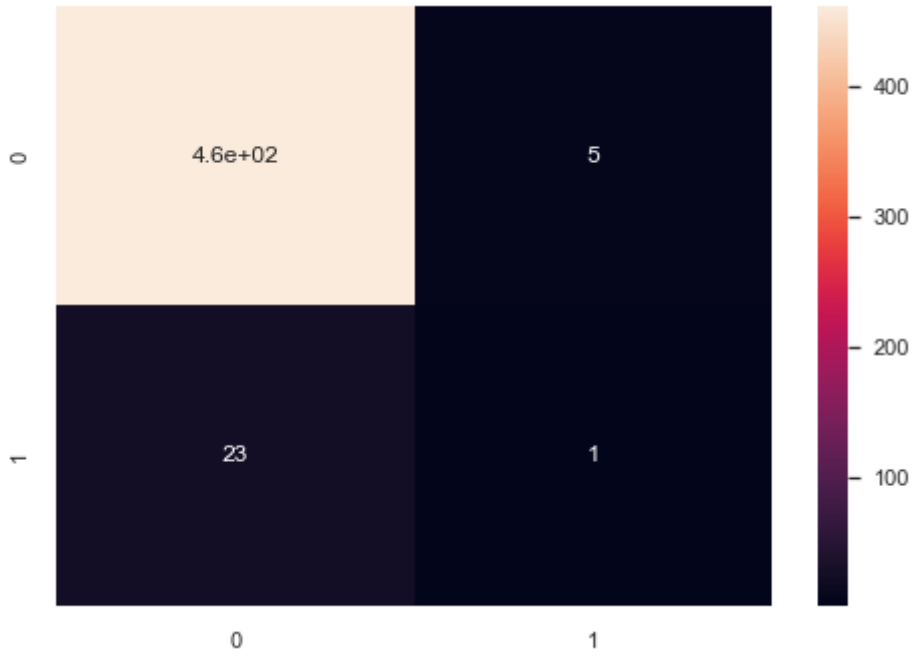
```
Accuracy on validation set for XGB model: 94.30%
```

In [48]:
```
print(classification_report(Y1_Val, Y1_pred_xg))
```

```
              precision    recall  f1-score   support

           0       0.95      0.99      0.97       467
           1       0.17      0.04      0.07        24

    accuracy                           0.94       491
   macro avg       0.56      0.52      0.52       491
weighted avg       0.91      0.94      0.93       491
```

In [49]:
```
#Confusion Matrix
sns.heatmap(confusion_matrix(Y1_Val, Y1_pred_xg),annot=True)
```

Out[49]:
```
<AxesSubplot:>
```



In [50]:
```
params = {
 'min_child_weight': [1, 5, 10, 15],
 'gamma': [0.5, 1, 1.5, 2, 5, 10, 15],
 'subsample': [0.6, 0.8, 1.0, 1,2],
 'colsample_bytree': [0.6, 0.8, 1.0],
 'max_depth': [3, 4, 5, 6, 8, 10],
 'eta': [0.001, 0.01, 0.1, 1, 10],
 'learning_rate': [0.001, 0.01, 0.1, 1, 10]
 }
```

In [51]:
```
#Hyper-Parameter Tuning
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.preprocessing import PowerTransformer
skf = RepeatedStratifiedKFold(n_splits=500, n_repeats=200, random_state = 1001)
rand_search_xg = RandomizedSearchCV(xgb_clf, param_distributions=params, n_iter=50
```

In [52]:
```
Data_transformed_xg = PowerTransformer().fit_transform(X1_Val)
rand_search_xg.fit(Data_transformed_xg, Y1_Val, eval_metric='logloss')
```

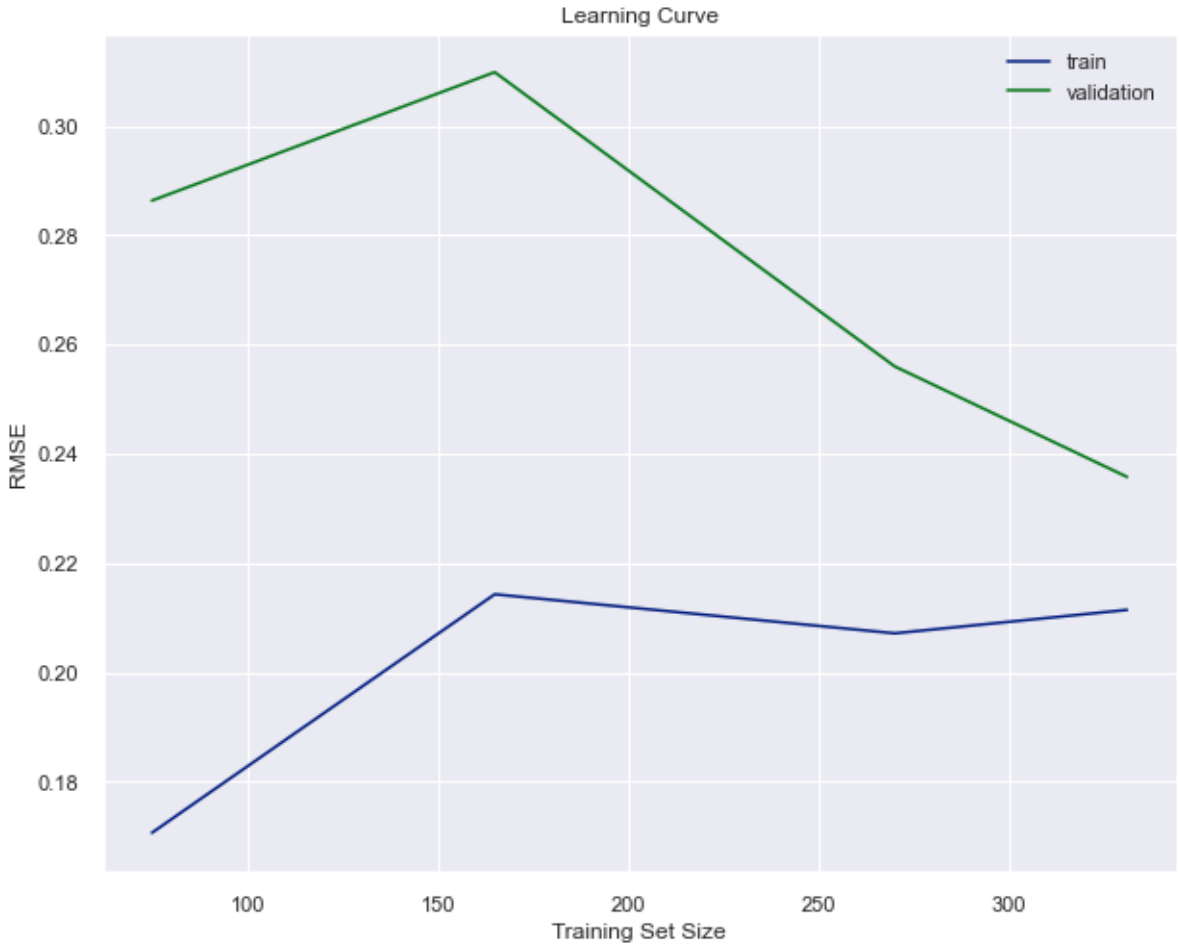
```
Fitting 5 folds for each of 50 candidates, totalling 250 fits
```

Out[52]:
```
RandomizedSearchCV(estimator=XGBClassifier(base_score=0.5, booster='gbtree',
                                           callbacks=None, colsample_bylevel=1,
                                           colsample_bynode=1,
                                           colsample_bytree=1,
                                           early_stopping_rounds=None,
                                           enable_categorical=False,
                                           eval_metric=None, feature_types=None,
                                           gamma=0, gpu_id=-1,
                                           grow_policy='depthwise',
                                           importance_type=None,
                                           interaction_constraints='',
                                           learning_rate=0.30000001...
                                           n_estimators=100, n_jobs=0,
                                           num_parallel_tree=1,
                                           predictor='auto', random_state=0, ...),
                   n_iter=50, n_jobs=-1,
                   param_distributions={'colsample_bytree': [0.6, 0.8, 1.0],
                                        'eta': [0.001, 0.01, 0.1, 1, 10],
                                        'gamma': [0.5, 1, 1.5, 2, 5, 10, 15],
                                        'learning_rate': [0.001, 0.01, 0.1, 1,
                                                          10],
                                        'max_depth': [3, 4, 5, 6, 8, 10],
                                        'min_child_weight': [1, 5, 10, 15],
                                        'subsample': [0.6, 0.8, 1.0, 1, 2]},
                   scoring='accuracy', verbose=3)
```

In [53]:
```
rand_search_xg.best_params_
```

Out[53]:
```
{'subsample': 0.6,
 'min_child_weight': 10,
 'max_depth': 3,
 'learning_rate': 10,
 'gamma': 10,
 'eta': 10,
 'colsample_bytree': 0.6}
```

In [54]:
```
#Learning Curve
xgb_clf_HP = XGBClassifier(subsample= 1, min_child_weight= 1,max_depth= 5,learning_
train_sizes, train_scores, test_scores = learning_curve(
    estimator=xgb_clf_HP,
    X=X1_train,
    y=Y1_train,
    cv=5,
    scoring="neg_root_mean_squared_error",
    train_sizes = [1, 75, 165, 270, 331]
)
train_mean = -train_scores.mean(axis=1)
test_mean = -test_scores.mean(axis=1)
plt.subplots(figsize=(10,8))
plt.plot(train_sizes, train_mean, label="train")
plt.plot(train_sizes, test_mean, label="validation")
plt.title("Learning Curve")
plt.xlabel("Training Set Size")
plt.ylabel("RMSE")
plt.legend(loc="best")
plt.show()
```



In [55]:
```
rand_search_xg.best_score_
```

Out[55]:
```
0.9511234796949083
```

Random Forest

In [56]:
```
from sklearn.ensemble import RandomForestClassifier
import pydot
from xgboost import XGBClassifier
import io
%matplotlib inline

clf_rf=RandomForestClassifier(n_estimators=100)
clf_rf.fit(X1_train,Y1_train)
y_pred_rf=clf_rf.predict(X1_Val)
print("Accuracy on validation set for Random Forest model: {:.2%}: ".format(metric
```

```
Accuracy on validation set for Random Forest model: 95.11%:
```

In [57]:
```
tree_small = clf_rf.estimators_[5]
export_graphviz(tree_small, out_file = 'small_tree.dot', feature_names = fn, round
(graph, ) = pydot.graph_from_dot_file('small_tree.dot')
graph.write_png('small_tree.png');
```

In [58]:
```
#Hyper-Parameter Tuning
param_grid = {
'n_estimators': [50, 100, 200, 300],
'min_samples_leaf': [1, 5, 10],
'max_depth': [2, 4, 6, 8, 10],
'max_features': ['auto', 'sqrt'],
'bootstrap': [True, False]}
```

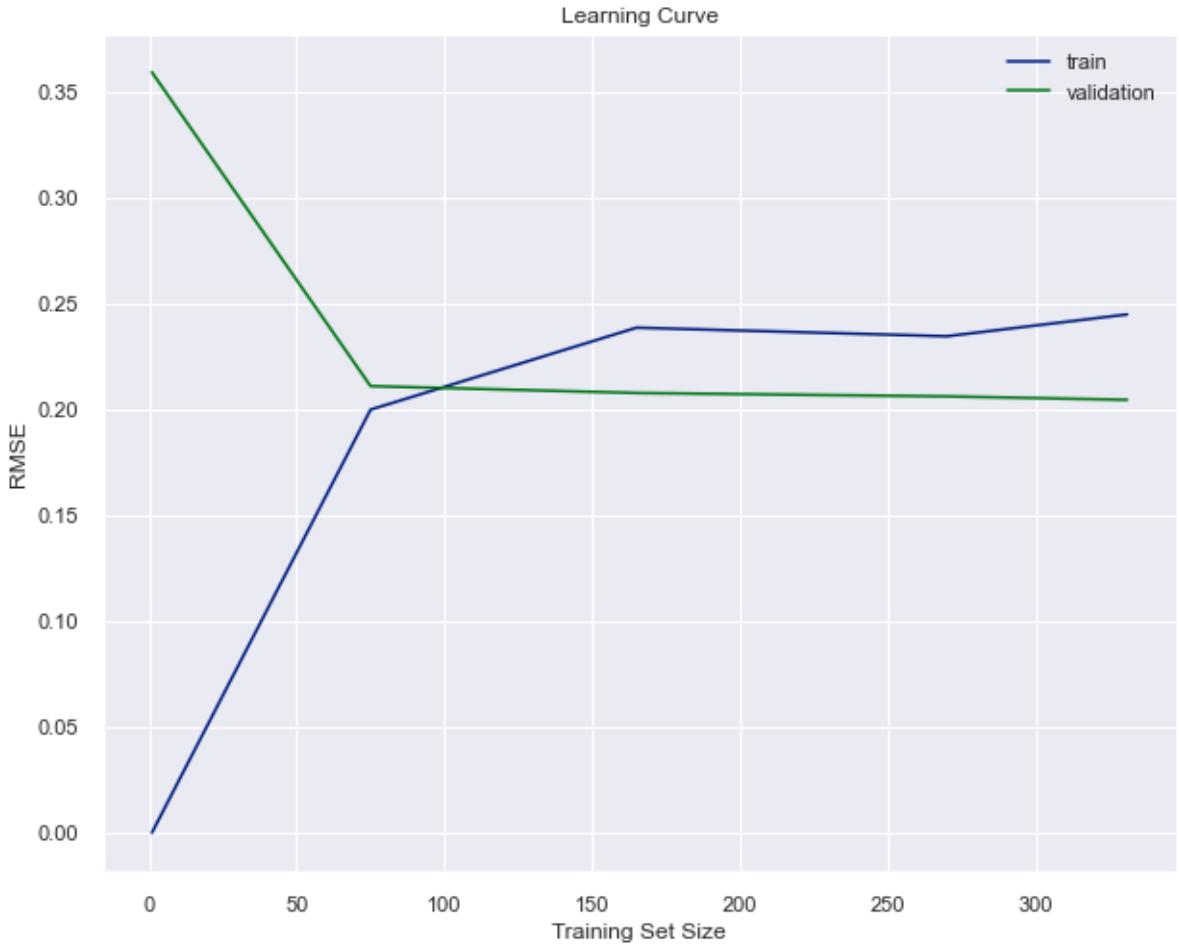
```
forest_1_HP = RandomizedSearchCV(clf_rf, param_dist)
forest_1_HP.fit(X1_train,Y1_train)
```

Out[58]:
```
RandomizedSearchCV(estimator=RandomForestClassifier(),
                   param_distributions={'max_depth': [3, None],
                                        'min_samples_leaf': <scipy.stats._distn_in
frastructure.rv_frozen object at 0x00000239C15F50A0>,
                                        'min_samples_split': <scipy.stats._distn_i
nfrastructure.rv_frozen object at 0x00000239C10FBD60>})
```

In [59]:
```
print("Tuned Random Forest Parameters: {}".format(forest_1_HP.best_params_))
print("Best score is {}".format(forest_1_HP.best_score_))
```

```
Tuned Random Forest Parameters: {'max_depth': None, 'min_samples_leaf': 8, 'min_sa
mples_split': 5}
Best score is 0.9581258359254494
```

In [60]:
```
#Learning Curve
forest_2_HP = RandomForestClassifier(max_depth= 3, min_samples_leaf= 1, min_samples
train_sizes, train_scores, test_scores = learning_curve(
 estimator=forest_2_HP,
 X=X1_train,
 y=Y1_train,
 cv=5,
 scoring="neg_root_mean_squared_error",
 train_sizes = [1, 75, 165, 270, 331]
)
train_mean = -train_scores.mean(axis=1)
test_mean = -test_scores.mean(axis=1)
plt.subplots(figsize=(10,8))
plt.plot(train_sizes, train_mean, label="train")
plt.plot(train_sizes, test_mean, label="validation")
plt.title("Learning Curve")
plt.xlabel("Training Set Size")
plt.ylabel("RMSE")
plt.legend(loc="best")
plt.show()
```

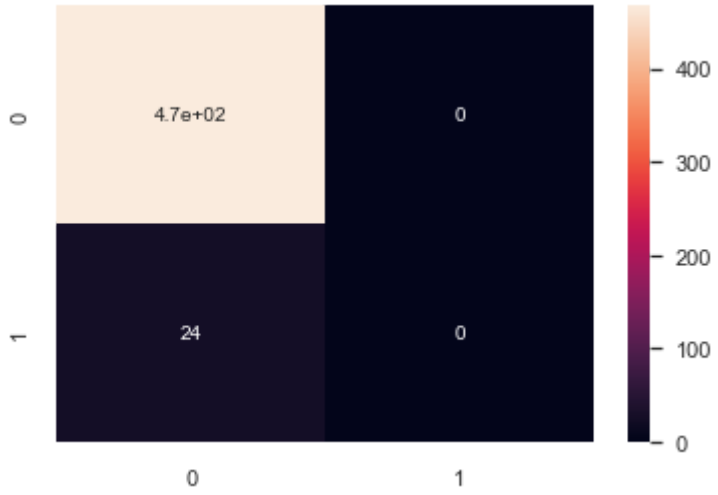


In [61]:
```
print(classification_report(Y1_Val, y_pred_rf))
```

```
              precision    recall  f1-score   support

           0       0.95      1.00      0.97       467
           1       0.00      0.00      0.00        24

    accuracy                           0.95       491
   macro avg       0.48      0.50      0.49       491
weighted avg       0.90      0.95      0.93       491
```

In [62]:
```
#Confusion Matrix

from sklearn.metrics import confusion_matrix

sns.heatmap(confusion_matrix(Y1_Val, y_pred_rf),annot=True)
```

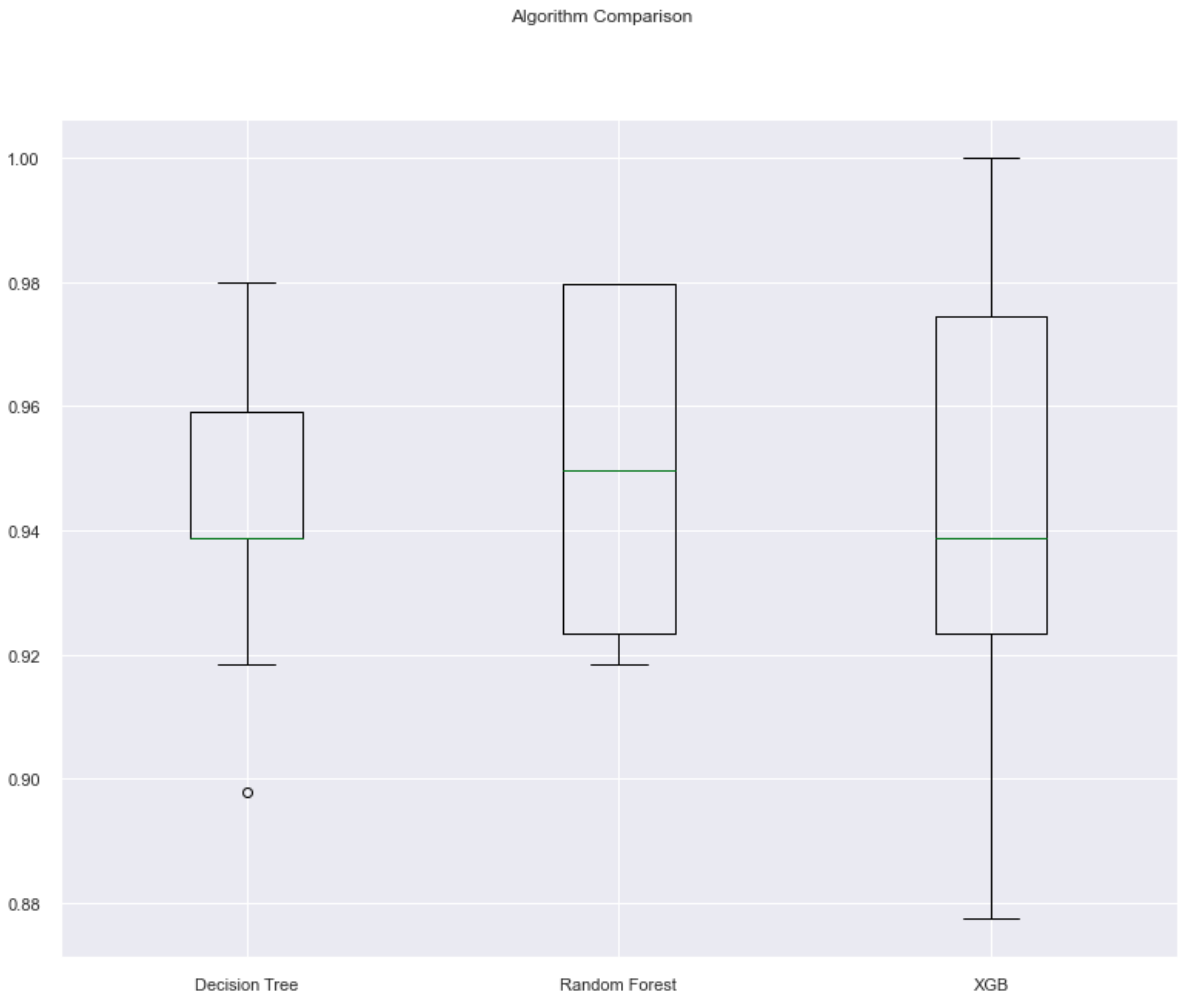
Out[62]:
```
<AxesSubplot:>
```



Model Comparision based on accuracy

In [63]:
```
from sklearn import model_selection
models = []

models.append(('Decision Tree', clf_2_HP))
models.append(('Random Forest', forest_2_HP))
models.append(('XGB', xgb_clf_HP))

results = []
names = []
scoring = 'accuracy'
for name, model in models:
 kfold = model_selection.KFold(n_splits=10, shuffle=True, random_state=None)
 cv_results = model_selection.cross_val_score(model, X1_Val, Y1_Val, cv=kfold, sco
 results.append(cv_results)
 names.append(name)
 msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
 print(msg)
# boxplot algorithm comparison
fig = plt.figure(figsize=(13, 10))
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
# plt.figure(figsize=(15, 15))
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

```
Decision Tree: 0.944939 (0.024292)
Random Forest: 0.951143 (0.026080)
XGB: 0.942857 (0.036278)
```



Statistical Significance - Paired t-test

In [64]:
```
from scipy import stats
from mlxtend.evaluate import paired_ttest_5x2cv

t, p = paired_ttest_5x2cv(estimator1=clf_2_HP,
estimator2=forest_2_HP,
X=X1_train, y=Y1_train,
 random_seed=1)
alpha = 0.05
print('t statistic: %.3f' % t)
print('p value: %.3f' % p)
if p > alpha:
 print("Failed to reject null hypotesis since p-value > 0.05")
else:
 print("We can reject null hypotesis since p-value < 0.05. Hence we can conclude th
```

```
t statistic: 0.000
p value: 1.000
Failed to reject null hypotesis since p-value > 0.05
```

In [65]:
```
from scipy import stats
from mlxtend.evaluate import paired_ttest_5x2cv

t, p = paired_ttest_5x2cv(estimator1=clf_2_HP,
estimator2=xgb_clf_HP,
X=X1_train, y=Y1_train,
 random_seed=1)
alpha = 0.05
print('t statistic: %.3f' % t)
print('p value: %.3f' % p)
```

```
if p > alpha:
 print("Failed to reject null hypotesis since p-value > 0.05")
else:
 print("We can reject null hypotesis since p-value < 0.05. Hence we can conclude th
```

```
t statistic: 1.007
p value: 0.360
Failed to reject null hypotesis since p-value > 0.05
```

In [66]:
```
from scipy import stats
from mlxtend.evaluate import paired_ttest_5x2cv

t, p = paired_ttest_5x2cv(estimator1=forest_2_HP,
estimator2=xgb_clf_HP,
X=X1_train, y=Y1_train,
 random_seed=1)
alpha = 0.05
print('t statistic: %.3f' % t)
print('p value: %.3f' % p)
if p > alpha:
 print("Failed to reject null hypotesis since p-value > 0.05")
else:
 print("We can reject null hypotesis since p-value < 0.05. Hence we can conclude th
```

```
t statistic: 1.355
p value: 0.233
Failed to reject null hypotesis since p-value > 0.05
```

In [ ]:

In [ ]:

In [ ]: