In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm

import seaborn as sns
%matplotlib inline
plt.rcParams['figure.figsize'] = (12,8)
dataFrame=pd.read_csv('healthcare-dataset-stroke-data.csv')
print(dataFrame)
```

```
         id  gender   age  hypertension  heart_disease ever_married  \
0      9046    Male  67.0             0              1          Yes
1     51676  Female  61.0             0              0          Yes
2     31112    Male  80.0             0              1          Yes
3     60182  Female  49.0             0              0          Yes
4      1665  Female  79.0             1              0          Yes
...     ...     ...   ...           ...            ...          ...
5105  18234  Female  80.0             1              0          Yes
5106  44873  Female  81.0             0              0          Yes
5107  19723  Female  35.0             0              0          Yes
5108  37544    Male  51.0             0              0          Yes
5109  44679  Female  44.0             0              0          Yes

          work_type Residence_type  avg_glucose_level   bmi   smoking_status
\
0           Private          Urban             228.69  36.6  formerly smoked
1     Self-employed          Rural             202.21   NaN     never smoked
2           Private          Rural             105.92  32.5     never smoked
3           Private          Urban             171.23  34.4           smokes
4     Self-employed          Rural             174.12  24.0     never smoked
...             ...            ...                ...   ...              ...
5105        Private          Urban              83.75   NaN     never smoked
5106  Self-employed          Urban             125.20  40.0     never smoked
5107  Self-employed          Rural              82.99  30.6     never smoked
5108        Private          Rural             166.29  25.6  formerly smoked
5109       Govt_job          Urban              85.28  26.2          Unknown

      stroke
0          1
1          1
2          1
3          1
4          1
...      ...
5105       0
5106       0
5107       0
5108       0
5109       0

[5110 rows x 12 columns]
```

In [15]:

```
dataFrame.describe()
```

Out[15]:

```
                 id          age  hypertension  heart_disease  avg_glucose_level          bmi
count   5110.000000  5110.000000   5110.000000    5110.000000        5110.000000  4909.000000
 mean  36517.829354    43.226614      0.097456       0.054012         106.147677    28.893237
  std  21161.721625    22.612647      0.296607       0.226063          45.283560     7.854067
  min     67.000000     0.080000      0.000000       0.000000          55.120000    10.300000
  25%  17741.250000    25.000000      0.000000       0.000000          77.245000    23.500000
  50%  36932.000000    45.000000      0.000000       0.000000          91.885000    28.100000
  75%  54682.000000    61.000000      0.000000       0.000000         114.090000    33.100000
  max  72940.000000    82.000000      1.000000       1.000000         271.740000    97.600000
```

In [4]:

```
df= dataFrame.drop_duplicates()
```

In [5]:

```
df.describe()
```

Out[5]:

```
                 id          age  hypertension  heart_disease  avg_glucose_level          bmi
count   5110.000000  5110.000000   5110.000000    5110.000000        5110.000000  4909.000000
 mean  36517.829354    43.226614      0.097456       0.054012         106.147677    28.893237
  std  21161.721625    22.612647      0.296607       0.226063          45.283560     7.854067
  min     67.000000     0.080000      0.000000       0.000000          55.120000    10.300000
  25%  17741.250000    25.000000      0.000000       0.000000          77.245000    23.500000
  50%  36932.000000    45.000000      0.000000       0.000000          91.885000    28.100000
  75%  54682.000000    61.000000      0.000000       0.000000         114.090000    33.100000
  max  72940.000000    82.000000      1.000000       1.000000         271.740000    97.600000
```

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
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 [7]:

```
df['bmi'].fillna(df['bmi'].median(), inplace=True)
```

In [8]:

```
df.isnull().sum()
```

Out[8]:

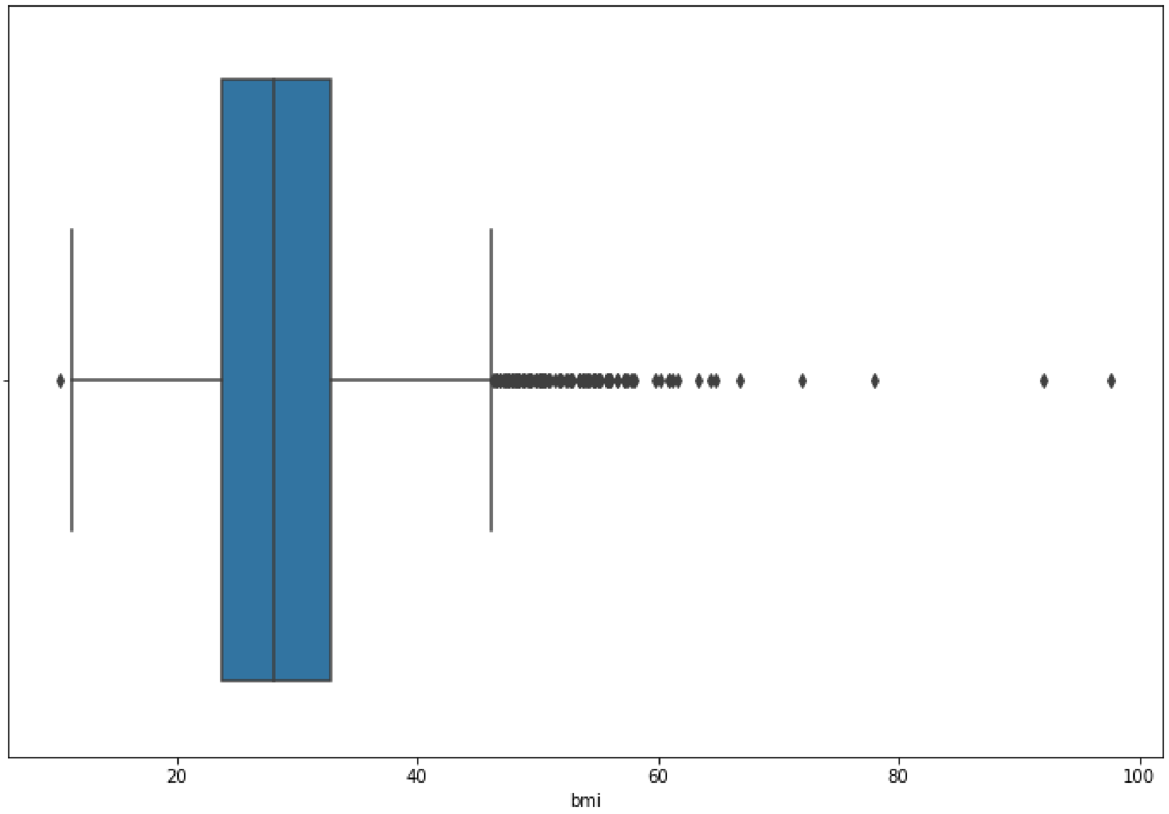
```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                  0
smoking_status       0
stroke               0
dtype: int64
```

In [9]:

```
sns.boxplot(x='bmi',data=df)
```

Out[9]:

```
<AxesSubplot:xlabel='bmi'>
```



In [9]:

```
q1=df.quantile(0.25)
q3=df.quantile(0.75)
IQR= q3-q1
print(q1)
print(q3)
print(IQR)
```

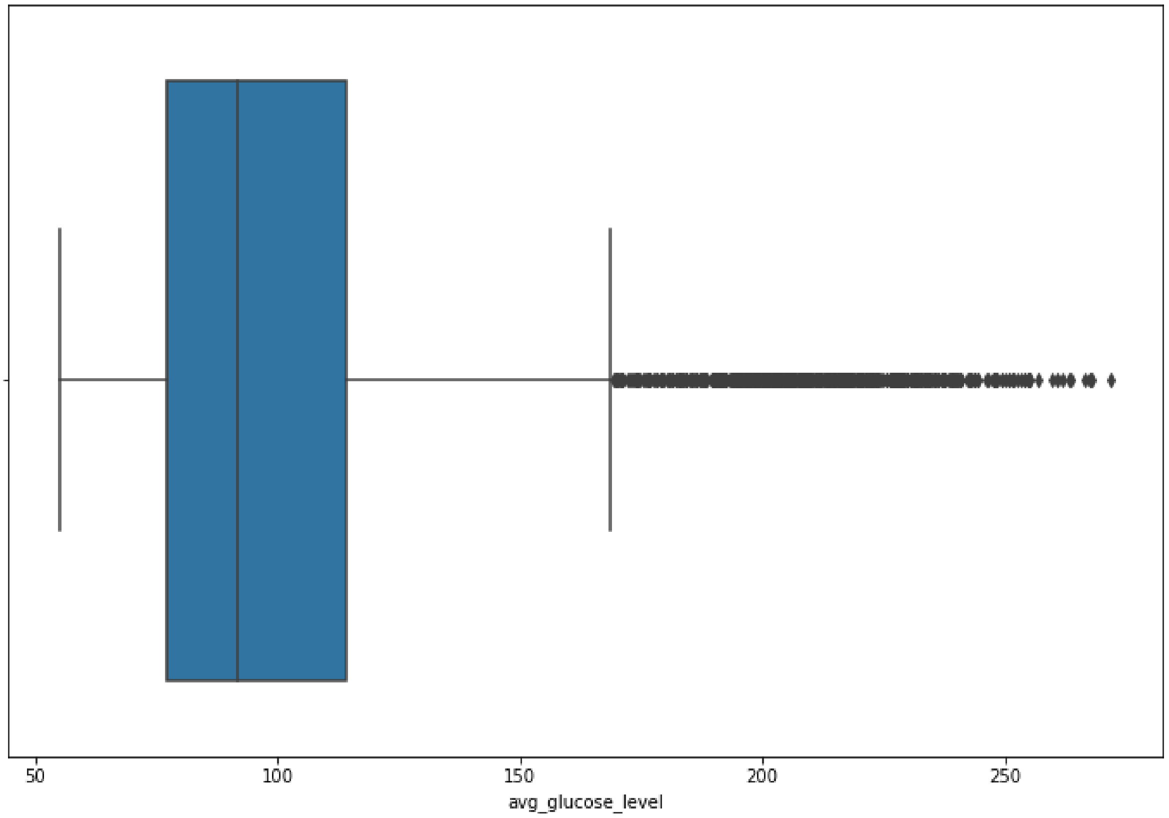
```
id                   17741.250
age                     25.000
hypertension             0.000
heart_disease            0.000
avg_glucose_level       77.245
bmi                     23.800
stroke                   0.000
Name: 0.25, dtype: float64
id                   54682.00
age                     61.00
hypertension             0.00
heart_disease            0.00
avg_glucose_level      114.09
bmi                     32.80
stroke                   0.00
Name: 0.75, dtype: float64
id                   36940.750
age                     36.000
hypertension             0.000
heart_disease            0.000
avg_glucose_level       36.845
bmi                      9.000
stroke                   0.000
dtype: float64
```

In [10]:

```
sns.boxplot(x='avg_glucose_level',data=df)
```

Out[10]:

```
<AxesSubplot:xlabel='avg_glucose_level'>
```

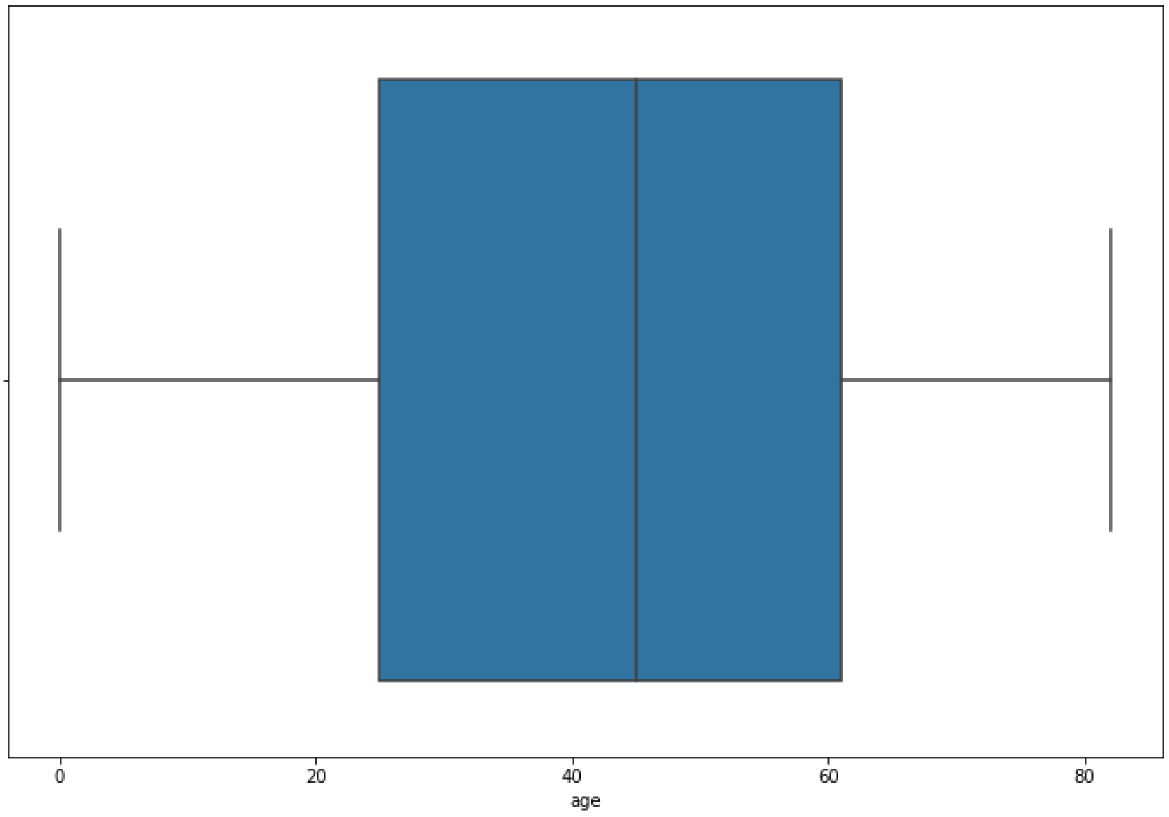


In [11]:

```
sns.boxplot(x='age',data=df)
```

Out[11]:

```
<AxesSubplot:xlabel='age'>
```



In [16]:

```
print(y1)
```

```
0       1
1       1
2       1
3       1
4       1
       ..
5105    0
5106    0
5107    0
5108    0
5109    0
Name: stroke, Length: 5110, dtype: int64
```

In [17]:

```
df['gender'] = df['gender'].replace({'Male':0,'Female':1,'Other':-1}).astype(np.uint8)
```

In [27]:

```
X1=df.drop(["id"], axis=1)
```

In [28]:

```
X1.head()
```

Out[28]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_gl
0	0	67.0	0	1	Yes	Private	Urban	
1	1	61.0	0	0	Yes	Self-employed	Rural	
2	0	80.0	0	1	Yes	Private	Rural	
3	1	49.0	0	0	Yes	Private	Urban	
4	1	79.0	1	0	Yes	Self-employed	Rural	

In [39]:

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import sklearn
from sklearn import svm
from sklearn.datasets import make_blobs

from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler,LabelEncoder
svm_pipeline = Pipeline(steps = [('scale',StandardScaler()),('SVM',SVC(random_state=42, pro
```

In [40]:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test =  train_test_split(X1, y1)
```

```
from imblearn.over_sampling import SMOTE

oversample = SMOTE()
X_train_resh, y_train_resh = oversample.fit_resample(X_train, y_train.ravel())
```

In [34]:

```
gender'] = df['gender'].replace({'Male':0,'Female':1,'Other':-1}).astype(np.uint8)
Residence_type'] = df['Residence_type'].replace({'Rural':0,'Urban':1}).astype(np.uint8)
work_type'] = df['work_type'].replace({'Private':0,'Self-employed':1,'Govt_job':2,'children'
ever_married'] = df['ever_married'].replace({'Yes':1, 'No':0}).astype(np.uint8)
smoking_status'] = df['smoking_status'].replace({'never smoked':0,'Unknown':1,'formerly smok
```

In [35]:

```
X1=df
```

In [36]:

```
X1.head()
```

Out[36]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type
0	9046	0	67.0	0	1	1	0	1
1	51676	1	61.0	0	0	1	1	0
2	31112	0	80.0	0	1	1	0	0
3	60182	1	49.0	0	0	1	0	1
4	1665	1	79.0	1	0	1	1	0

In [41]:

```
from imblearn.over_sampling import SMOTE

oversample = SMOTE()
X_train_resh, y_train_resh = oversample.fit_resample(X_train, y_train.ravel())
```

In [45]:

```
from sklearn.model_selection import train_test_split,cross_val_score
svm_cv = cross_val_score(svm_pipeline,X_train_resh,y_train_resh,cv=10,scoring='f1')
svm_cv.mean()
```

Out[45]:

```
0.8749045394396754
```

In [46]:

```
from sklearn.metrics import confusion_matrix

svm_pipeline.fit(X_train_resh,y_train_resh);
svm_train_predict = svm_pipeline.predict(X_train)
svm_pred = svm_pipeline.predict(X_test)
svm_cm = confusion_matrix(y_train,svm_train_predict)
svm_cm
```

Out[46]:

```
array([[3096,  547],
       [  71,  118]], dtype=int64)
```

In [14]:

```

```

In [ ]:

```

```