In [4]:
```
import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (10,10)
```

In [5]:
```
data = pd.read_csv(r"C:\Users\krish\Downloads\healthcare-dataset-stroke-data.csv")
data
```

Out[5]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoki
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	former
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	nev
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	nev
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	nev
...	...	...	...	...	...	...	...	...	...	...	
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	nev
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	nev
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	nev
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	former
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	

5110 rows × 12 columns

In [6]:
```
data.describe()
```

Out[6]:

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000	5110.000000
mean	36517.829354	43.226614	0.097456	0.054012	106.147677	28.893237	0.048728
std	21161.721625	22.612647	0.296607	0.226063	45.283560	7.854067	0.215320
min	67.000000	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	17741.250000	25.000000	0.000000	0.000000	77.245000	23.500000	0.000000
50%	36932.000000	45.000000	0.000000	0.000000	91.885000	28.100000	0.000000
75%	54682.000000	61.000000	0.000000	0.000000	114.090000	33.100000	0.000000
max	72940.000000	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

finding missing value

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
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 [8]:
```
data['bmi'].describe()
```

Out[8]:
```
count    4909.000000
mean       28.893237
std         7.854067
min        10.300000
25%        23.500000
50%        28.100000
75%        33.100000
max        97.600000
Name: bmi, dtype: float64
```

In [9]:
```
data['bmi'].fillna(data['bmi'].mean(), inplace=True)     # Filling the null value
```

In [10]:
```
data.isnull().sum()
```

Out[10]:
```
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 [11]:
```
data.drop('id', axis=1, inplace=True)
```

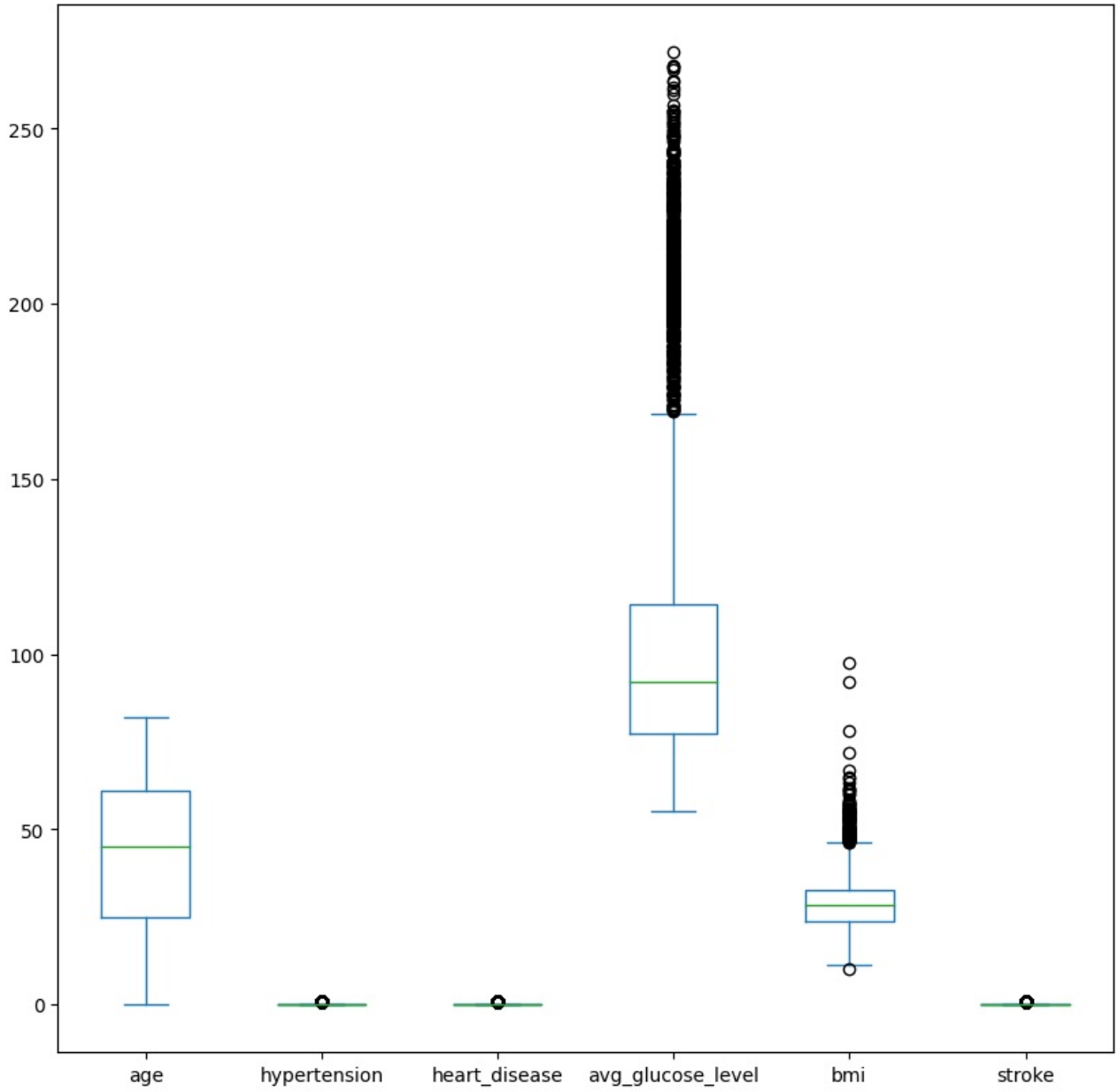
In [12]:
```
data
```

Out[12]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking
0	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly
1	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	28.893237	never
2	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	never
3	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	
4	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	never
...	...	...	...	...	...	...	...	...	...	
5105	Female	80.0	1	0	Yes	Private	Urban	83.75	28.893237	never
5106	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.000000	never
5107	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.600000	never
5108	Male	51.0	0	0	Yes	Private	Rural	166.29	25.600000	formerly
5109	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.200000	U

5110 rows × 11 columns

In [13]:
```
data.plot(kind = 'box')
plt.show()
```



In [14]:
```
data['avg_glucose_level'].describe()     #  finding Outlier
```

Out[14]:
```
count    5110.000000
mean      106.147677
std        45.283560
min        55.120000
25%        77.245000
50%        91.885000
75%       114.090000
max       271.740000
Name: avg_glucose_level, dtype: float64
```

In [15]:
```
data[data['avg_glucose_level']>114.09]
```

Out[15]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking
0	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly
1	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	28.893237	never
3	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	
4	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	never
5	Male	81.0	0	0	Yes	Private	Urban	186.21	29.000000	formerly
...	...	...	...	...	...	...	...	...	...	
5071	Male	81.0	0	0	Yes	Private	Rural	135.32	35.800000	U
5076	Female	34.0	0	0	Yes	Private	Rural	174.37	23.000000	never
5086	Female	51.0	0	0	Yes	Private	Urban	152.56	21.800000	U
5106	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.000000	never
5108	Male	51.0	0	0	Yes	Private	Rural	166.29	25.600000	formerly

1277 rows × 11 columns

Conversion of data into interger form

In [16]:
```
from sklearn.preprocessing import LabelEncoder
enc = LabelEncoder()
```

In [51]:
```
gender = enc.fit_transform(data['gender'])
smoking_status = enc.fit_transform(data['smoking_status'])
work_type = enc.fit_transform(data['work_type'])
ever_married = enc.fit_transform(data['ever_married'])
residence = enc.fit_transform(data['Residence_type'])
```

In [52]:
```
data['gender'] = gender
data['smoking_status'] = smoking_status
data['work_type'] = work_type
data['ever_married'] = ever_married
data['Residence_type'] = residence
```

In [53]:
```
data
```

Out[53]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking
0	1	67.0	0	1	1	2	1	228.69	36.600000	
1	0	61.0	0	0	1	3	0	202.21	28.893237	
2	1	80.0	0	1	1	2	0	105.92	32.500000	
3	0	49.0	0	0	1	2	1	171.23	34.400000	
4	0	79.0	1	0	1	3	0	174.12	24.000000	
...	...	...	...	...	...	...	...	...	...	
5105	0	80.0	1	0	1	2	1	83.75	28.893237	
5106	0	81.0	0	0	1	3	1	125.20	40.000000	
5107	0	35.0	0	0	1	3	0	82.99	30.600000	
5108	1	51.0	0	0	1	2	0	166.29	25.600000	
5109	0	44.0	0	0	1	0	1	85.28	26.200000	

5110 rows × 11 columns

Train and testing dataset

In [54]:
```
X = data.drop('stroke' , axis=1)
Y=data['stroke']
```

In [55]:
```
X.head()
```

Out[55]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_sta
0	1	67.0	0	1	1	2	1	228.69	36.600000	
1	0	61.0	0	0	1	3	0	202.21	28.893237	
2	1	80.0	0	1	1	2	0	105.92	32.500000	
3	0	49.0	0	0	1	2	1	171.23	34.400000	
4	0	79.0	1	0	1	3	0	174.12	24.000000	

In [34]:
```
Y
```

Out[34]:
```
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 [56]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test=train_test_split(X,Y,test_size=0.2,random_state=10)
```

In [57]:
```
X_train
```

Out[57]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking
2285	1	49.0	0	0	1	2	0	79.64	28.893237	
4733	1	67.0	0	0	1	2	0	83.16	25.500000	
3905	1	78.0	0	0	1	2	1	208.85	24.400000	
4700	1	47.0	0	0	1	2	0	110.14	30.500000	
4939	0	59.0	0	0	1	2	1	71.08	28.100000	
...	...	...	...	...	...	...	...	...	...	...
1180	0	62.0	0	0	1	2	0	82.57	36.000000	
3441	0	59.0	0	0	1	3	1	90.06	28.900000	
1344	1	47.0	0	0	1	2	0	86.37	39.200000	
4623	1	25.0	0	0	1	0	1	166.38	23.100000	
1289	0	80.0	0	0	1	3	0	72.61	27.600000	

4088 rows × 10 columns

In [47]:
```
Y_train
```

Out[47]:
```
2285    0
4733    0
3905    0
4700    0
4939    0
       ..
1180    0
3441    0
1344    0
4623    0
1289    0
Name: stroke, Length: 4088, dtype: int64
```

In [48]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   gender             5110 non-null   int64  
 1   age                5110 non-null   float64
 2   hypertension       5110 non-null   int64  
 3   heart_disease      5110 non-null   int64  
 4   ever_married       5110 non-null   int32  
 5   work_type          5110 non-null   int64  
 6   Residence_type     5110 non-null   object 
 7   avg_glucose_level  5110 non-null   float64
 8   bmi                5110 non-null   float64
 9   smoking_status     5110 non-null   int64  
 10  stroke             5110 non-null   int64  
dtypes: float64(3), int32(1), int64(6), object(1)
memory usage: 419.3+ KB
```

Normalization

In [58]:
```
from sklearn.preprocessing import StandardScaler
std=StandardScaler()
```

In [60]:
```
X_train_std=std.fit_transform(X_train)
X_test_std=std.transform(X_test)
```

In [61]:
```
X_test_std
```

Out[61]:
```
array([[-0.83780372,  0.64952452, -0.33069968, ..., -0.12678509,
         1.38727506,  1.51158251],
       [ 1.19359699,  0.60537571, -0.33069968, ..., -0.35586361,
         0.12078063, -1.28365994],
       [ 1.19359699,  0.95856622, -0.33069968, ..., -0.83414241,
         0.00238781, -0.35191245],
       ...,
       [-0.83780372,  0.87026859, -0.33069968, ..., -1.08555387,
         1.17836876,  0.57983503],
       [ 1.19359699,  0.60537571, -0.33069968, ..., -0.66056457,
         0.32968693, -0.35191245],
       [-0.83780372, -1.29302329, -0.33069968, ..., -0.75962556,
        -1.31545016, -1.28365994]])
```

In [62]:
```
X_test_std
```

Out[62]:
```
array([[-0.83780372,  0.64952452, -0.33069968, ..., -0.12678509,
         1.38727506,  1.51158251],
       [ 1.19359699,  0.60537571, -0.33069968, ..., -0.35586361,
         0.12078063, -1.28365994],
       [ 1.19359699,  0.95856622, -0.33069968, ..., -0.83414241,
         0.00238781, -0.35191245],
       ...,
       [-0.83780372,  0.87026859, -0.33069968, ..., -1.08555387,
         1.17836876,  0.57983503],
       [ 1.19359699,  0.60537571, -0.33069968, ..., -0.66056457,
         0.32968693, -0.35191245],
       [-0.83780372, -1.29302329, -0.33069968, ..., -0.75962556,
        -1.31545016, -1.28365994]])
```

Training Model

In [ ]:
```

```

Decision Tree

In [63]:
```
from sklearn.tree import  DecisionTreeClassifier
dt = DecisionTreeClassifier()
```

In [64]:
```
dt.fit(X_train_std, Y_train)
```

Out[64]:
```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

In [65]:
```
dt.feature_importances_
```

Out[65]:
```
array([0.02702864, 0.17199553, 0.01289764, 0.03028217, 0.03424253,
       0.04617912, 0.0512278 , 0.28478321, 0.27325123, 0.06811212])
```

In [67]:
```
X_train.columns
```

Out[67]:
```
Index(['gender', 'age', 'hypertension', 'heart_disease', 'ever_married',
       'work_type', 'Residence_type', 'avg_glucose_level', 'bmi',
       'smoking_status'],
      dtype='object')
```

In [72]:
```
Y_pred =dt.predict(X_test_std)
```

In [69]:
```
X_test
```

Out[69]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smokin
2413	0	58.00	0	0	1	2	0	100.42	39.500000	
1141	1	57.00	0	0	1	2	0	90.06	29.800000	
146	1	65.00	0	0	1	3	1	68.43	28.893237	
3883	0	1.64	0	0	0	4	1	69.89	18.100000	
1044	0	79.00	0	0	1	0	1	93.89	30.400000	
...	...	...	...	...	...	...	...	...	...	
2261	1	59.00	0	0	1	2	1	60.35	25.900000	
4712	1	57.00	0	0	1	2	1	93.04	29.200000	
4971	0	63.00	0	0	1	2	1	57.06	37.900000	
2224	1	57.00	0	0	1	2	0	76.28	31.400000	
4825	0	14.00	0	0	0	4	1	71.80	18.800000	

1022 rows × 10 columns

In [70]:
```
Y_test
```

Out[70]:
```
2413    0
1141    0
146     1
3883    0
1044    0
       ..
2261    0
4712    0
4971    0
2224    0
4825    0
Name: stroke, Length: 1022, dtype: int64
```

In [71]:
```
from sklearn.metrics import accuracy_score
```

In [76]:
```
 ac_dt = accuracy_score(Y_test,Y_pred)
```

In [77]:
```
ac_dt
```

Out[77]:
```
0.9031311154598826
```

Logistic Regression

In [78]:
```
from sklearn.linear_model import LogisticRegression
lg = LogisticRegression()
```

In [79]:
```
lg.fit(X_train_std,Y_train)
lg
```

Out[79]:
```
▾ LogisticRegression
LogisticRegression()
```

In [93]:
```
y_pred = lg.predict(X_test_std)
y_pred
```

Out[93]:
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

In [96]:
```
y_train
```

Out[96]:
```
4450    0
4472    0
138     1
1943    0
4811    0
       ..
1175    0
4805    0
3852    0
2734    0
4574    0
Name: stroke, Length: 4088, dtype: int64
```

In [97]:
```
x_test
```

Out[97]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking
4854	0	36.0	0	0	No	2	Rural	66.55	32.800000	
1867	0	55.0	0	0	Yes	2	Rural	63.47	27.800000	
4582	0	76.0	0	0	Yes	0	Urban	223.64	27.100000	
3048	0	62.0	1	0	Yes	3	Urban	75.78	28.893237	
263	0	40.0	0	0	Yes	2	Rural	95.04	42.400000	
...	...	...	...	...	...	...	...	...	...	...
2983	0	31.0	0	0	Yes	2	Rural	69.26	21.800000	
1030	1	56.0	0	0	Yes	2	Rural	156.18	25.300000	
782	0	38.0	1	0	Yes	3	Urban	91.00	33.300000	
3330	0	56.0	0	0	Yes	2	Urban	80.08	25.600000	
1465	0	21.0	0	0	No	2	Rural	121.11	21.000000	

1022 rows × 10 columns

In [94]:
```
accuracy_score(Y_test,y_pred)
```

Out[94]:
```
0.9383561643835616
```

In [ ]:
```

```