Introduction:

The goal of this project is to use classification models to predict whether a patient is likely to get a stroke based on the input parameters like gender, age, various diseases, and smoking status. Each row in the data provides relevant information about the patient.

Design:

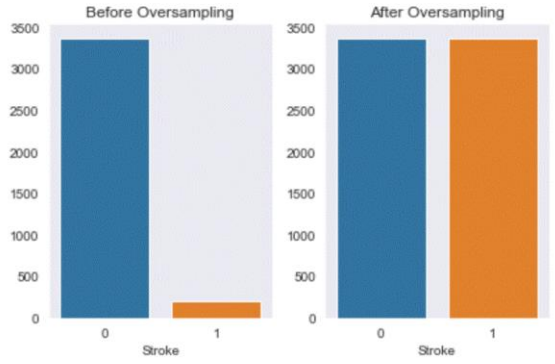
I choose this dataset to assist people who is likely to have a stroke based on information they share, healthcare sector, and doctors to aware people to prevent stroke by know the cause of it.

Data:

The dataset from Kaggle (Stroke Prediction Dataset). The dataset contains 61332 data point with 5110 rows and 12 columns. The Dataset contains both categorical and numerical features. Only 'bmi' feature having some null values. I can see that this dataset is an imbalanced dataset since the number of patients that are likely to get a stroke is smaller when compared with the number of patients that did not.

Algorithms:

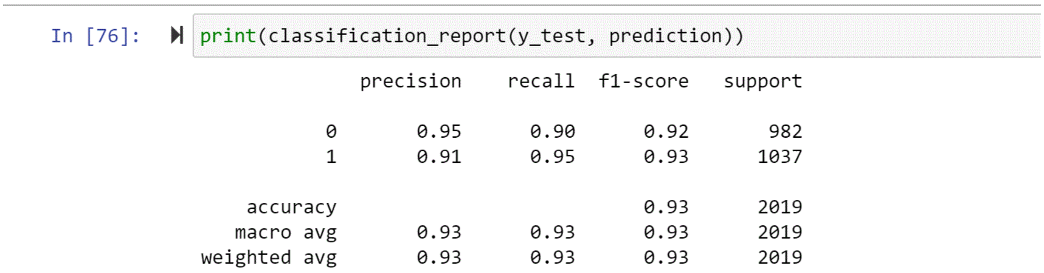
- Since our dataset is highly imbalanced, there is a risk that our models will be biased toward predicting no stroke. To combat this issue To make it balanced we use a technique called SMOTE (Synthetic Minority Oversampling Technique). This technique SMOTE increases number of sample of minority classes by linear interpolation. After applying SMOTE to the training set, the stroke vs. no-stroke rows is more balanced.



Preparing the Data for Prediction

1. Converting the Categorical Columns into Numerical by Mapping each category to an integer value using map() on pandas series object
2. Spliting the Data in Training and Testing Samples

- I want to predict if patient will have a stroke by applying Random Forest Classifier
  - Accuracy Score: 92.72%



```
In [76]:  print(classification_report(y_test, prediction))

                   precision    recall  f1-score   support

               0       0.95      0.90      0.92       982
               1       0.91      0.95      0.93      1037

        accuracy                           0.93      2019
       macro avg       0.93      0.93      0.93      2019
    weighted avg       0.93      0.93      0.93      2019
```

Tools:

- # To prevent the annoying warning from scikit learn package

import warnings

warnings.filterwarnings('ignore')

- # For suppressing warnings

warnings.filterwarnings("ignore")

- #import the essental libraries

import numpy as np data manipulation

import pandas as pd data manipulation

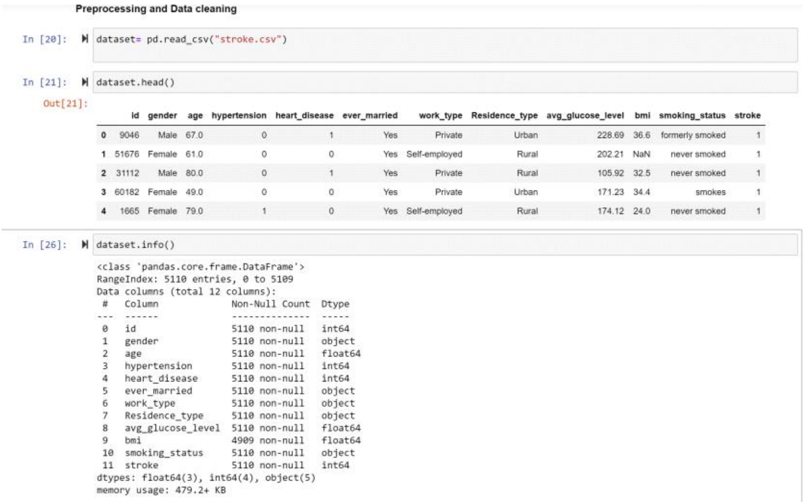
import seaborn as sns For visualization.

from matplotlib import pyplot as plt For visualization.

- PowerPoint to present my slides.

Communication:

1. Importing the Data using Pandas read_csv(). And calling head() and info() on the DataFrame



Preprocessing and Data cleaning

```
In [20]:  dataset= pd.read_csv("stroke.csv")
```

```
In [21]:  dataset.head()
```

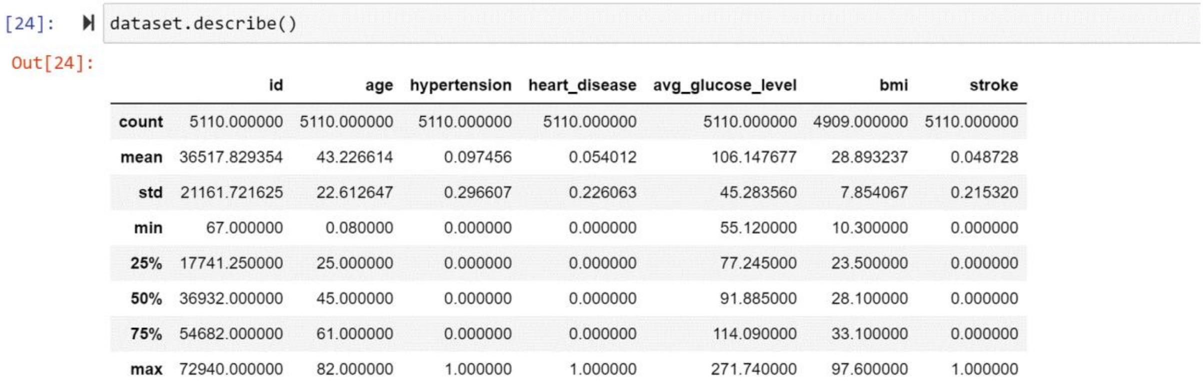
Out[21]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

```
In [26]:  dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 5110 non-null   int64  
 1   gender             5110 non-null   object 
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64  
 4   heart_disease      5110 non-null   int64  
 5   ever_married       5110 non-null   object 
 6   work_type          5110 non-null   object 
 7   Residence_type     5110 non-null   object 
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object 
 11  stroke             5110 non-null   int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

2. Just by looking at the sample of the dataset, we can figure out the columns and the type of data that they contain.



```
[24]:  dataset.describe()
```

Out[24]:

	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

Observation:

- The id column is a unique identifier.
- The dataset contains both categorical and numerical columns.

Categorical columns:

- **gender**: Gender of the patient.
- **hypertension**: whether the patient suffers from hypertension (1) or not (0).
- **heart_disease**: whether the patient suffers from heart disease (1) or not (0).
- **ever_married**: marital status of the patient if married (Yes) else (No).
- **work_type**: The type of occupation of the patient.
- **Resident_Type**: The type of residence of the patient.
- **smoking_status:** How often does the patient smoke (if ever).

Numerical columns:

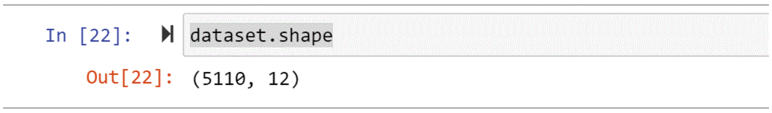
- **age**: Age of the Patient
- **avg_glucose_level**: Average Glucose Level of the patient.

- **bmi**: body mass index of the patient.

Output Column:

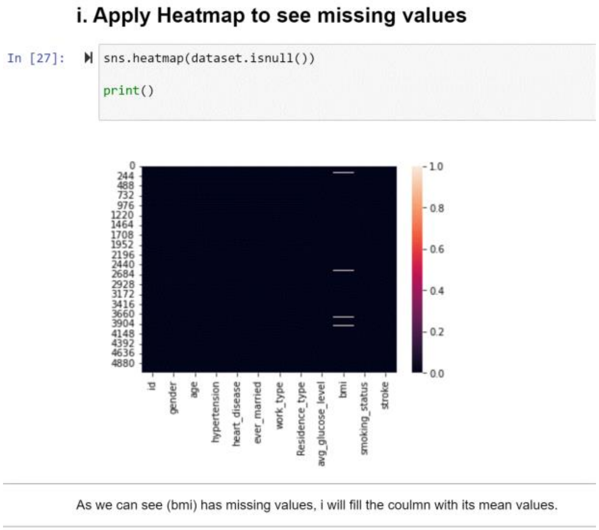
- **Stroke**: Whether the patient is likely to get a stroke (1) or not (0).

3. Get the idea of the size of data points by printing its shape.



```
In [22]:  dataset.shape
Out[22]:  (5110, 12)
```

4. Taking care of NA values

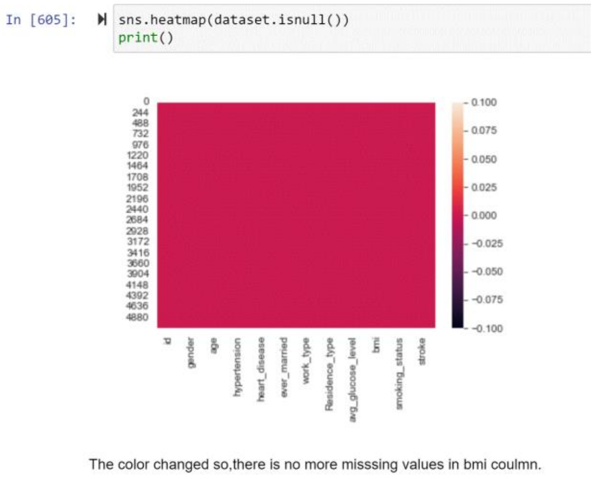


i. Apply Heatmap to see missing values

```
In [27]:  sns.heatmap(dataset.isnull())

          print()
```

As we can see (bmi) has missing values, i will fill the coulmn with its mean values.



```
In [605]:  sns.heatmap(dataset.isnull())
           print()
```

The color changed so,there is no more misssing values in bmi coulmn.

References:

- https://www.victorchang.edu.au/stroke?gclid=CjwKCAiAtdGNBhAmEiwAWxGcUlhJSxRn90QVjOfw8CZkFp4mMBlo0BHdnxRJMc2r0qyFA8lqN4Xr2RoCkGsQAvD_BwE

- https://stroke.org.sa/understand-stroke/

- https://www.pulseuniform.com/coffee-time/awareness-ribbons-guide-colors-and-meanings/

- https://stackabuse.com/python-dictionary-tutorial/