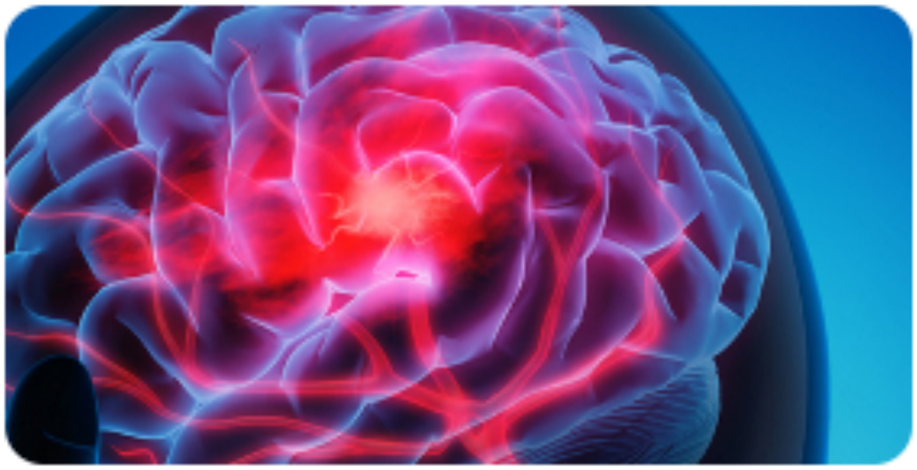
King Saud University
College of Computer and Information Sciences
Software Engineering department

SW 485: Selected topics in software engineering

Stroke Prediction Analysis Project Report

Phase #1

Group:4



Name	ID
Jouri Alanazi	441200780
Renad Alsubaie	441200868
Monerah Alsubaie	441200880

Supervised BY: Dr.Manal Bin Khonain

Contents

1. Introduction:

Machine learning is a subfield of artificial intelligence that involves the use of algorithms and statistical models to analyze and interpret complex data. In recent years, machine learning has become increasingly popular in the healthcare industry, particularly in the field of stroke prediction analysis.

Motivation:

A stroke is a serious medical condition that occurs when blood flow to the brain is interrupted, either due to a blockage in the blood vessels or bleeding in the brain. Early detection of stroke is crucial for timely treatment, which can help prevent serious complications and even death.

Machine learning algorithms can be used to analyze large datasets of patient information to identify patterns and risk factors that may contribute to stroke. By analyzing factors such as age, sex, medical history, and lifestyle habits, machine learning models can predict a patient's risk of stroke and help healthcare providers develop personalized treatment plans.

Objective and Expected Results:

The objective of using machine learning in stroke prediction analysis is to improve the accuracy and efficiency of stroke risk assessment. By analyzing large amounts of patient data using machine learning algorithms, healthcare providers can identify patterns and risk factors that may not be immediately apparent through traditional methods.

The expected results of using machine learning in stroke prediction analysis include:

1. Improved accuracy: Machine learning algorithms can identify subtle patterns in patient data that may not be immediately apparent through traditional methods. This can lead to more accurate stroke risk assessments and better treatment decisions.
2. Early detection: Machine learning models can be trained to detect early warning signs of stroke, allowing for timely interventions that can prevent serious complications.
3. Personalized treatment: By analyzing patient data, machine learning algorithms can help healthcare providers develop personalized treatment plans based on individual risk factors and medical history.
4. Reduced healthcare costs: Early detection and personalized treatment can help reduce healthcare costs by preventing costly complications and hospitalizations.

Overall, the use of machine learning in stroke prediction analysis has the potential to improve patient outcomes and advance medical research in the field of stroke prevention and treatment.

2. Machine learning Tasks:

In stroke prediction analysis, machine learning can be used for several tasks, including:

1. Feature selection: Machine learning algorithms can identify the most relevant features or risk factors that contribute to stroke prediction. This involves selecting the most important variables from a large dataset of patient information that can be used to develop a predictive model.

2. Classification: Machine learning algorithms can be used to classify patients into different risk categories based on their individual risk factors. This allows healthcare providers to identify patients who are at a higher risk of stroke and develop personalized prevention plans accordingly.

3. Regression: Machine learning algorithms can also be used to predict the probability of stroke occurrence or the time to stroke occurrence for individual patients. This can help healthcare providers develop more accurate and personalized treatment plans.

4. Clustering: Machine learning algorithms can group patients with similar risk profiles together, which can help healthcare providers identify common risk factors and develop more effective prevention strategies.

5. Anomaly detection: Machine learning algorithms can be used to identify patients with unusual or unexpected risk factors that may not be immediately apparent through traditional methods. This can help healthcare providers identify patients who may be at a higher risk of stroke and develop personalized prevention plans accordingly.

Overall, machine learning can be a powerful tool for stroke prediction analysis, allowing healthcare providers to develop more accurate and personalized prevention and treatment plans for individual patients.

3. Data:

1.Data Kind:

contains information about patients who may or may not have experienced a stroke. The dataset includes demographic information such as age, gender, and smoking status, as well as medical history such as hypertension, heart disease, and glucose levels. The target variable is whether or not the patient has had a stroke.
The dataset contains 5110 observations and 12 variables, This dataset can be used for modeling and predicting the likelihood of a patient having a stroke based on their demographic and medical information. The dataset is useful for researchers and healthcare professionals interested in developing stroke prevention and treatment strategies.

2.Data Source:

The Stroke Prediction Dataset available on Kaggle was collected and made available .The data was collected from patients who visited a healthcare facility in a rural area of India. The information was gathered through medical examinations and interviews with the patients.
We got the dataset from: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

3.Data Exploration:

Number of observations:

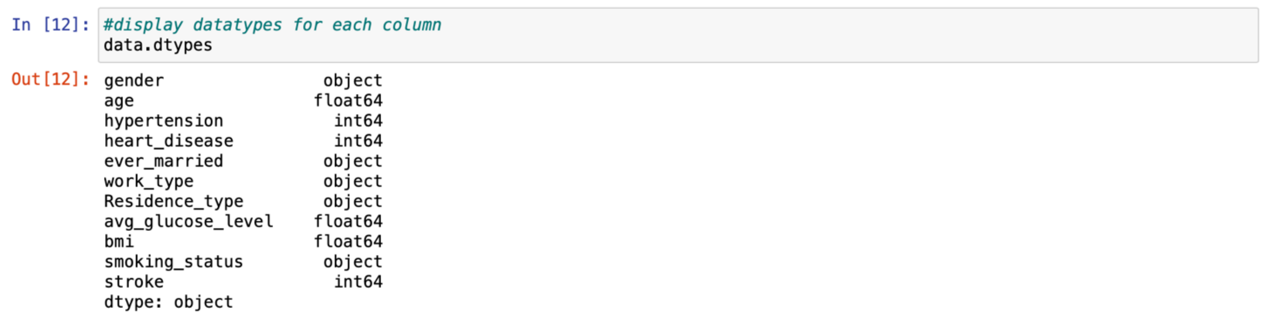
The Stroke Prediction Dataset contains 5110 observations or rows. Each row represents a patient in the dataset.

Number of variables and data type:

The Stroke Prediction Dataset contains 12 variables or columns. Here is the list of variables and their data types:

1. id: Numeric - unique identifier for each patient
2. gender: Categorical - male or female
3. age: Numeric - age of the patient in years
4. hypertension: Categorical - 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5. heart disease: Categorical - 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6. ever married: Categorical - "No" or "Yes"
7. work type: Categorical - "children", "Govt jov", "Never worked", "Private" or "Self-employed"
8. Residence type: Categorical - "Rural" or "Urban"
9. avg glucose level: Numeric - average glucose level in the patient's blood
10. bmi: Numeric - body mass index of the patient
11. smoking status: Categorical - "formerly smoked", "never smoked", "smokes" or "Unknown"
12. stroke: Categorical - 1 if the patient had a stroke, 0 if not

This Figure displays the data type for each variable in our dataset method:

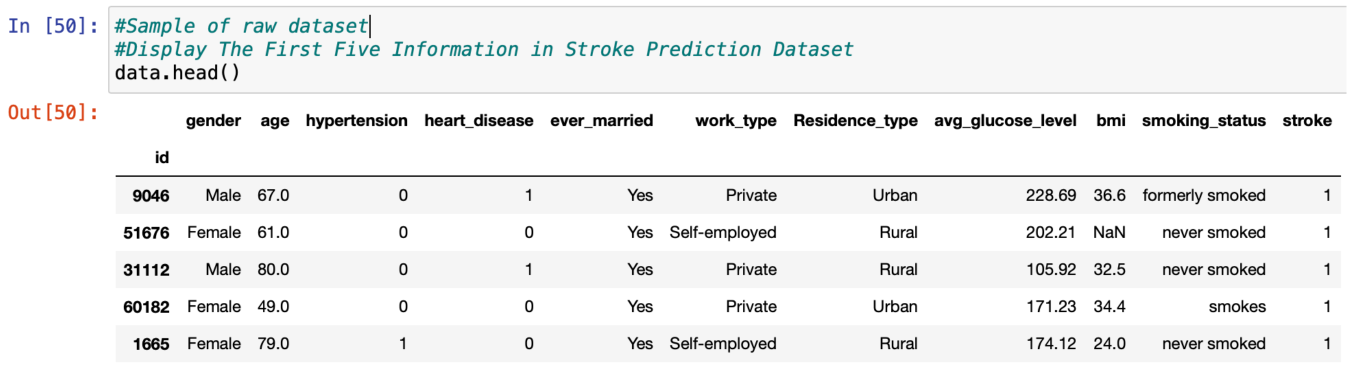


```
In [12]: #display datatypes for each column
         data.dtypes

Out[12]: gender                object
         age                  float64
         hypertension           int64
         heart_disease          int64
         ever_married          object
         work_type             object
         Residence_type        object
         avg_glucose_level    float64
         bmi                  float64
         smoking_status        object
         stroke                 int64
         dtype: object
```

Figure 1. Variables and their data types in the dataset

Sample of raw dataset:

To display a sample from the dataset, we also used the. head () [1] method. The first five rows of the dataset were displayed by the function.



```
In [50]: #Sample of raw dataset
         #Display The First Five Information in Stroke Prediction Dataset
         data.head()
```

Out[50]:

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

Figure 2.Raw Sample from a dataset

Displaying Graphs in Dataset:

```
In [115]: #Corrleation and we descripe it with grapg and if it's became more brighter then there is a relationship between the
          plt.figure(figsize=(20, 10))
          sns.heatmap(data.corr(numeric_only = True) , annot=True , cmap="Blues" , linecolor="White")

Out[115]: <Axes: >
```

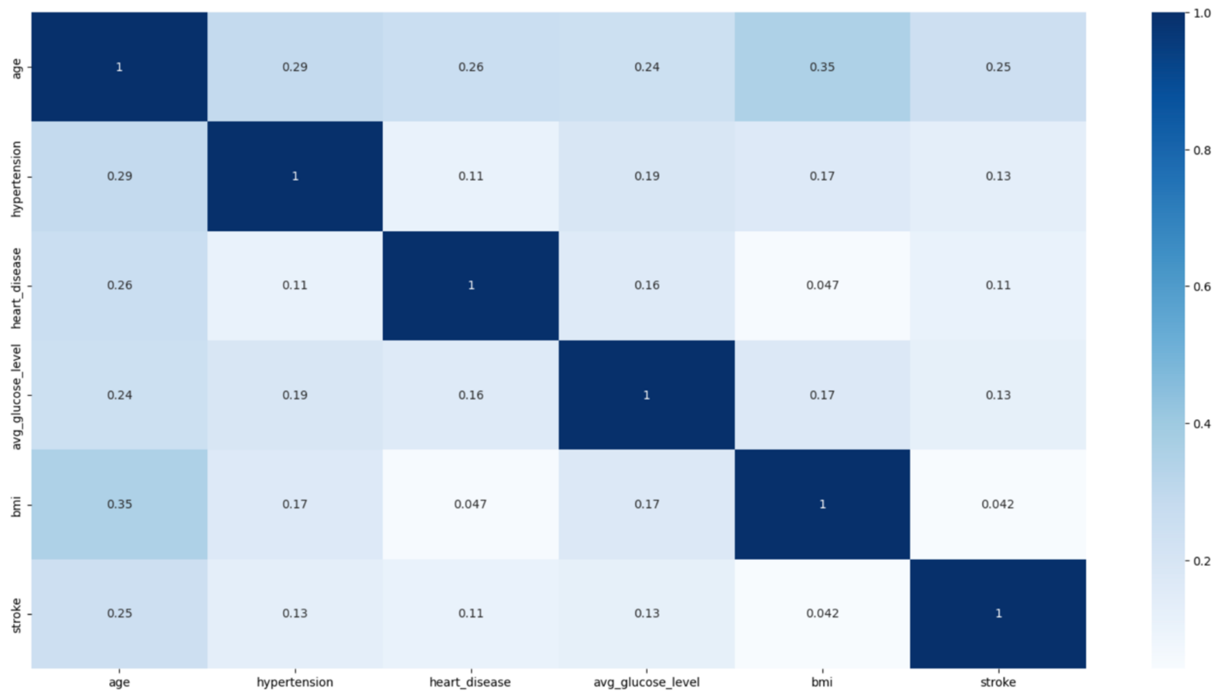


Figure 3. Correlation for the dataset

Variables Distribution

Distribution plot

This distribution plot depicts the overall distribution and shows the impact of age and avg_glucose_level and BMI.

```
In [52]: #Variables Distribution - Figure 1
         #Histogram Density
         column_list = ['age', 'avg_glucose_level', 'bmi']

         fig, ax = plt.subplots(1, 3, figsize=(12,6))

         for i, col in enumerate(column_list):
           sns.histplot(data=data, x=col, stat='density', hue='stroke', multiple='stack', ax=ax[i])
           sns.kdeplot(data=data, x=col, linewidth=3, hue='stroke', fill=False, multiple='stack', ax=ax[i])

         plt.tight_layout()
         plt.show()
```

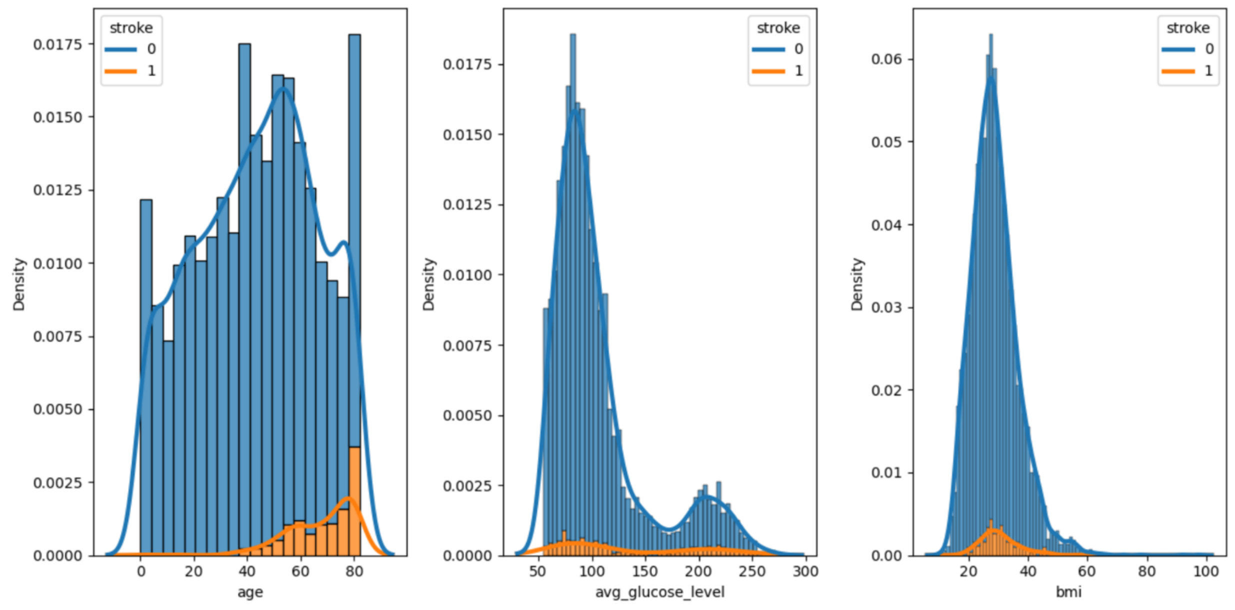


Figure 4.Distribution plot

Scatter plot

Scatter plot is usually used to see the correlation between attributes we want to know the correlation between patients age and average glucose level.

```
In [51]: #Variables Distribution – Figure 2
         #Scatter Plot
         plt.figure(figsize=(12,6))
         sns.jointplot(data=data, x='age', y='avg_glucose_level')
         plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```

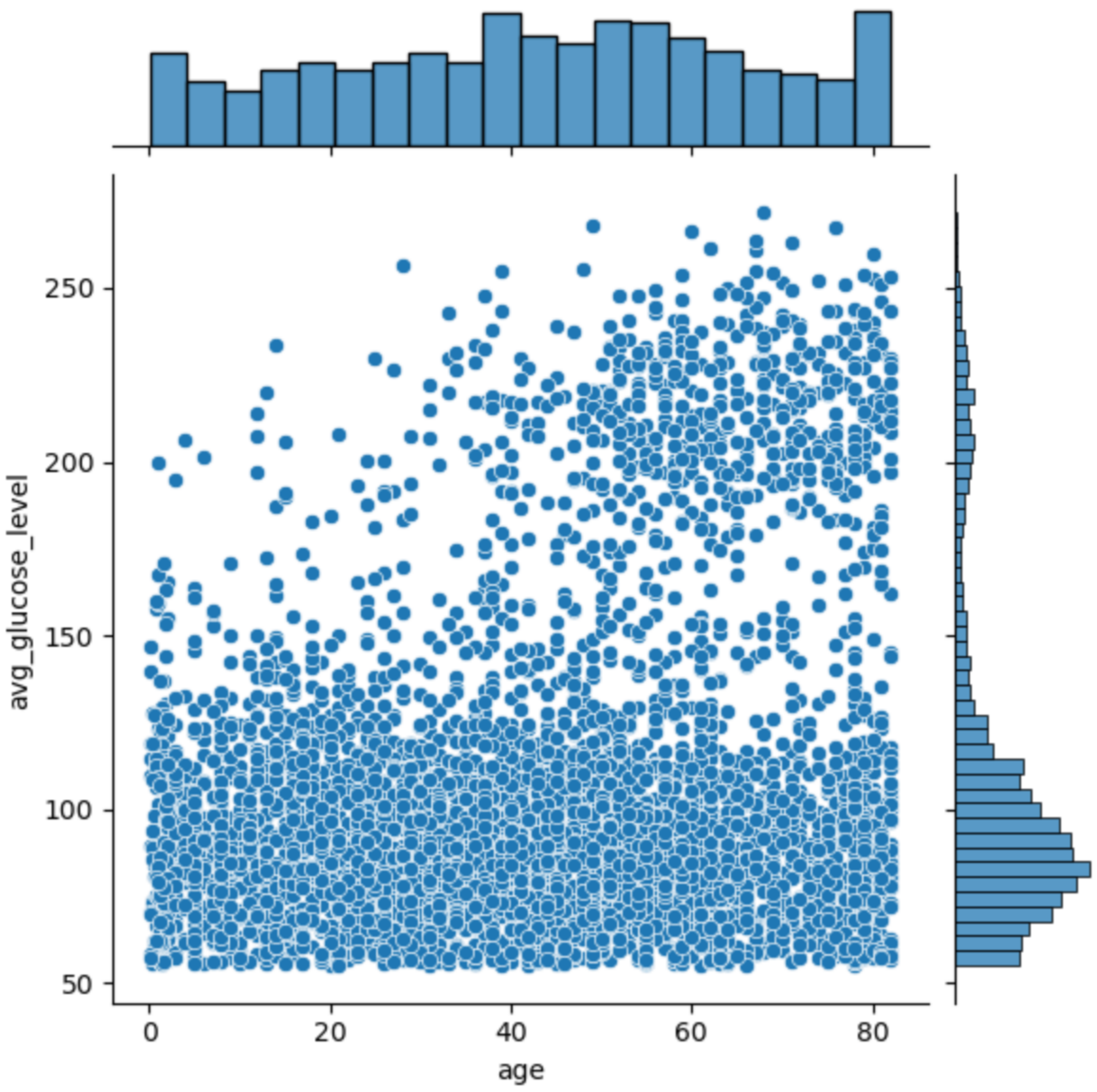


Figure 5.Joint plot

Pie Chart

Another way to see the portion of categories is using a pie chart. But, we must be very careful when using this chart type.

```
In [82]: #Variables Distribution - Figure 4
         # Create a pie chart
         plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%', startangle=90)
         plt.axis('equal')
         plt.title('Stroke vs No Stroke')
         plt.show()
```

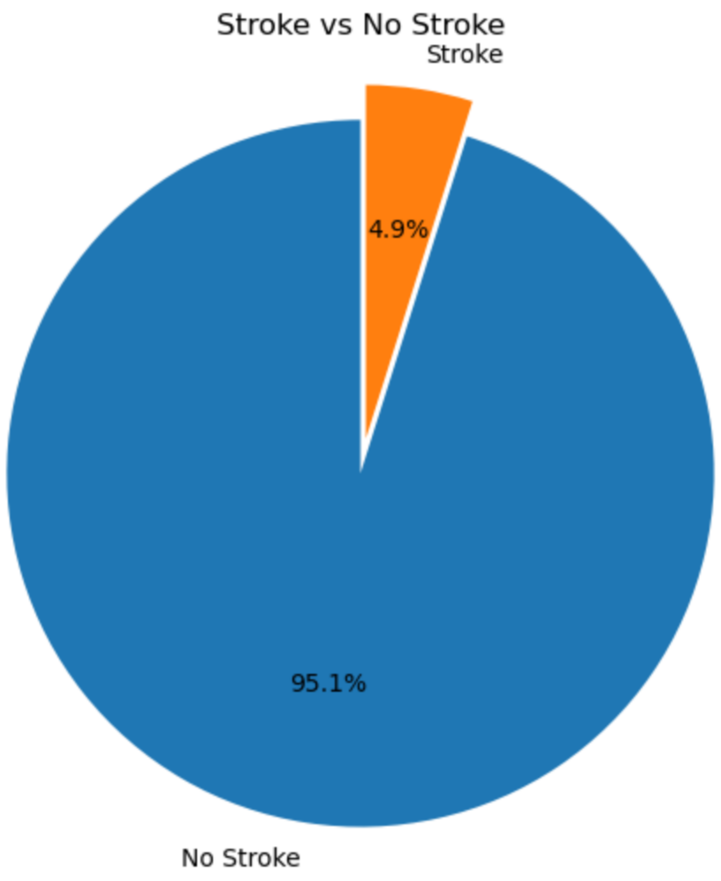


Figure 6.Pie Chart

Bar Plot

The Bar chart compares between the two type of gender (Female & Male).

```
In [53]: #Variables Distribution – Figure 3
         #Bar Plot
         data['gender'].value_counts().plot(kind='bar')
         print(data['gender'].value_counts())
         plt.title('Distribution of gender', fontsize=15)

         Female    2994
         Male      2115
         Other        1
         Name: gender, dtype: int64

Out[53]: Text(0.5, 1.0, 'Distribution of gender')
```

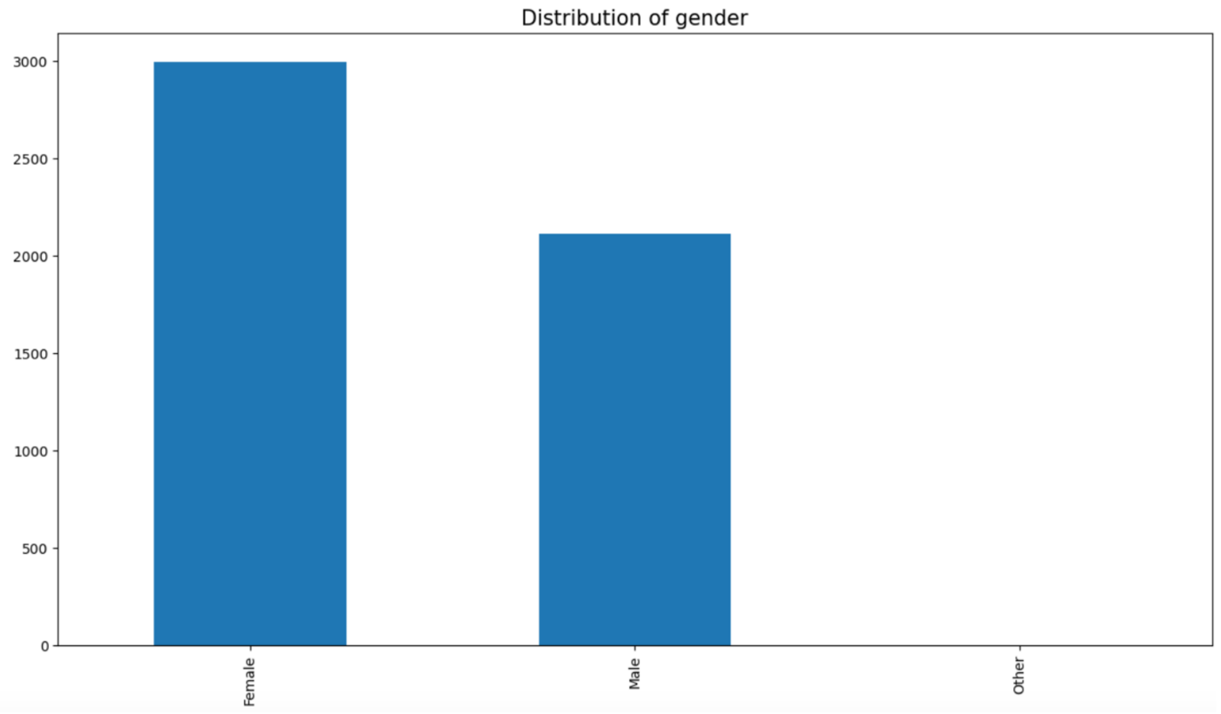
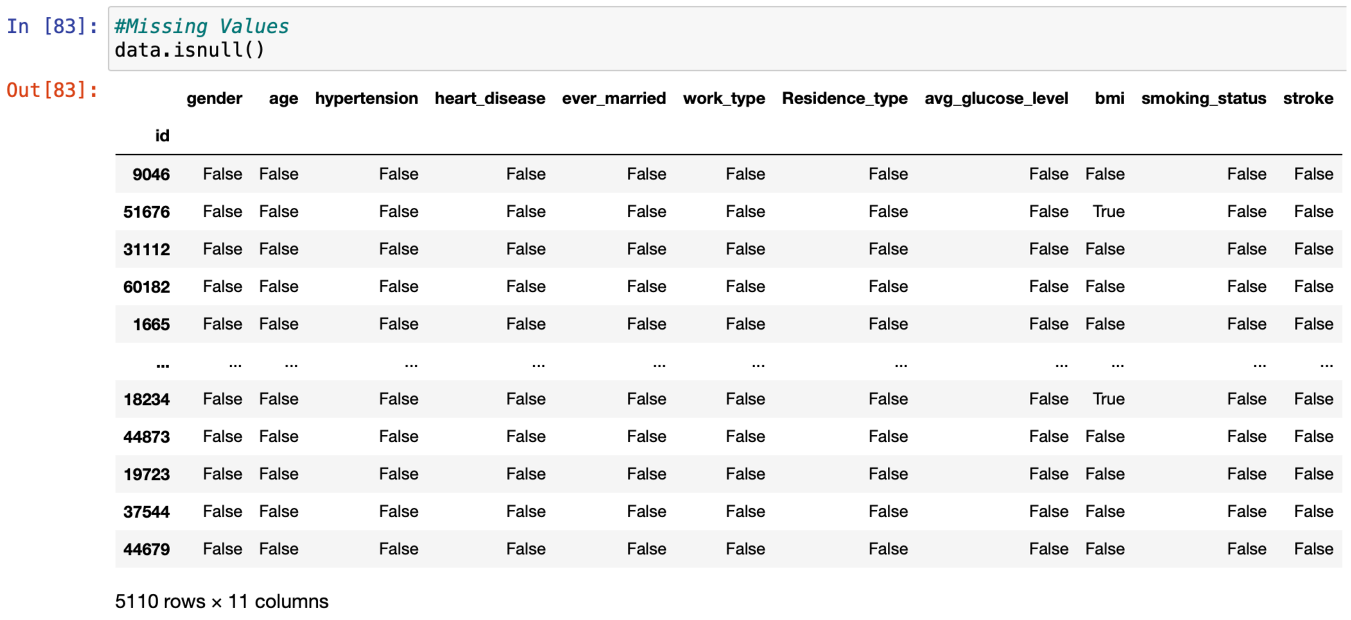


Figure 7.Bar Plot

Missing values(null):

To Detect any missing values we use. isnull()[1] function from our dataset:



```
In [83]: #Missing Values
         data.isnull()
```

Out[83]:

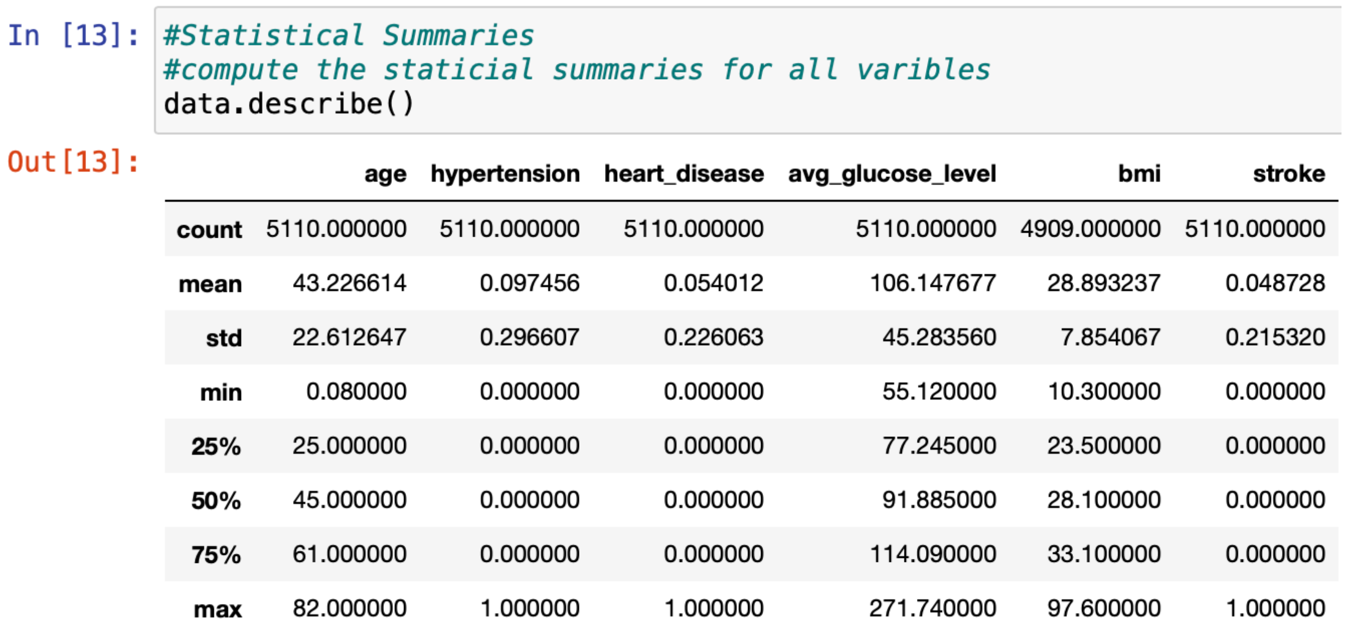
id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
9046	False	False	False	False	False	False	False	False	False	False	False
51676	False	False	False	False	False	False	False	False	True	False	False
31112	False	False	False	False	False	False	False	False	False	False	False
60182	False	False	False	False	False	False	False	False	False	False	False
1665	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
18234	False	False	False	False	False	False	False	False	True	False	False
44873	False	False	False	False	False	False	False	False	False	False	False
19723	False	False	False	False	False	False	False	False	False	False	False
37544	False	False	False	False	False	False	False	False	False	False	False
44679	False	False	False	False	False	False	False	False	False	False	False

5110 rows × 11 columns

Figure 8. Display Null Values

Statistical Summaries:

In this table we calculate mean, std, min, max for the numeric columns we will use. describe**()[1]** function from our dataset:



```
In [13]: #Statistical Summaries
         #compute the staticial summaries for all varibles
         data.describe()
```

Out[13]:

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

Figure 9. Display Statistical Summaries

4. Data preprocessing

The Stroke Prediction Dataset contains medical and demographic data for over 5000 patients, with the goal of predicting whether a patient is at risk of having a stroke based on their medical and demographic information.

Before we can model the data, it's important to preprocess it to ensure its quality and suitability for analysis. The preprocessing steps for this dataset may include:

1. Handling missing values: The dataset contains some missing values represented by "N/A" or "Unknown". These missing values can be handled by either imputing the missing values using statistical methods or dropping the rows with missing values if the missing values are too many.

2. Encoding categorical variables: The categorical variables in the dataset such as gender, ever_married, work_type, Residence_type, and smoking_status need to be encoded as numeric values to be used in modeling. This can be done using methods such as one-hot encoding or label encoding.

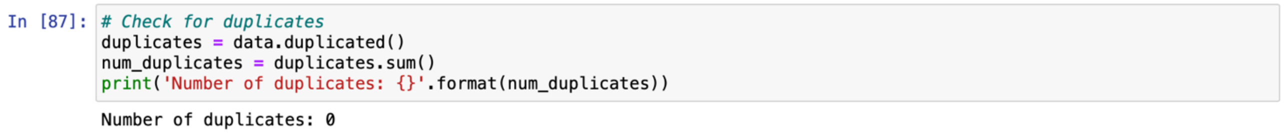
3. Scaling numeric variables: The numeric variables in the dataset such as age, avg_glucose_level, and bmi may need to be scaled to have a similar range of values. This can be done using methods such as standardization or normalization.

4. Handling imbalanced data: The dataset is imbalanced with a majority of patients not having a stroke. This can lead to biased models. Techniques such as oversampling the minority class or undersampling the majority class can be used to balance the data.

5. Feature selection: Not all variables in the dataset may be relevant for modeling. Feature selection techniques can be used to select the most important variables for predicting stroke.

By preprocessing the data, we can improve the quality of the data and make it suitable for modeling. We can then use various machine learning algorithms to predict stroke risk in patients and help healthcare providers make informed decisions.

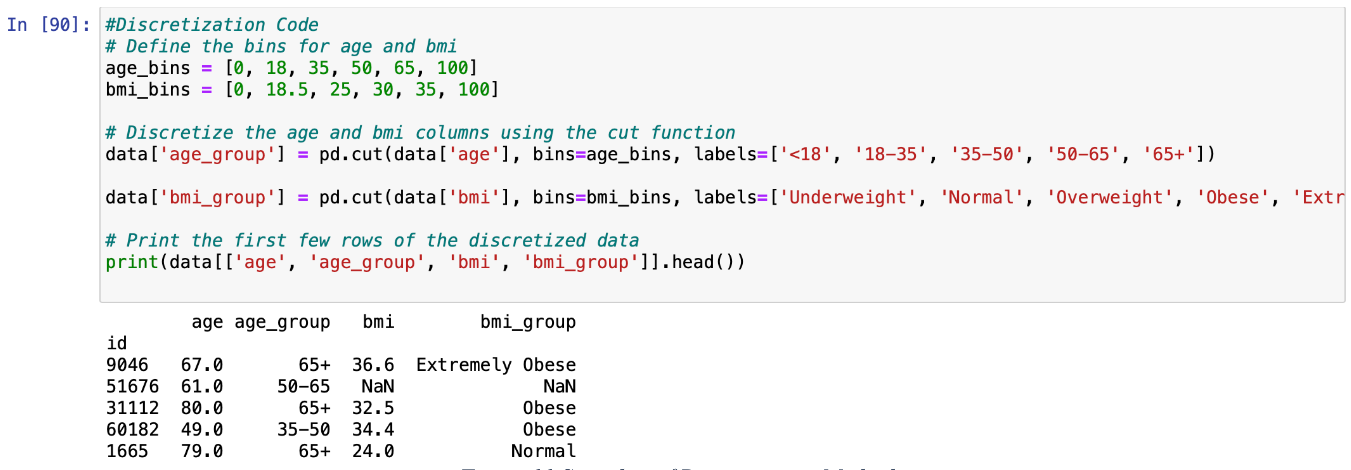
And we check if there are any duplicate values in our dataset and we did not find and below the snapshot.



```
In [87]: # Check for duplicates
         duplicates = data.duplicated()
         num_duplicates = duplicates.sum()
         print('Number of duplicates: {}'.format(num_duplicates))

         Number of duplicates: 0
```

Figure 10. Snapshot to check if there is duplicate data

Also, we make discretization for our dataset between age and BMI.

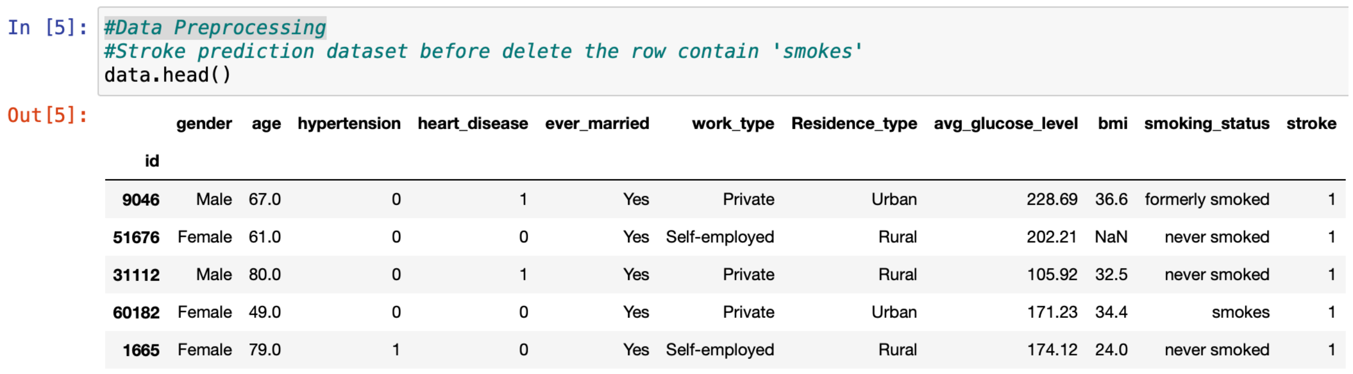


```
In [90]: #Discretization Code
         # Define the bins for age and bmi
         age_bins = [0, 18, 35, 50, 65, 100]
         bmi_bins = [0, 18.5, 25, 30, 35, 100]

         # Discretize the age and bmi columns using the cut function
         data['age_group'] = pd.cut(data['age'], bins=age_bins, labels=['<18', '18-35', '35-50', '50-65', '65+'])

         data['bmi_group'] = pd.cut(data['bmi'], bins=bmi_bins, labels=['Underweight', 'Normal', 'Overweight', 'Obese', 'Extr

         # Print the first few rows of the discretized data
         print(data[['age', 'age_group', 'bmi', 'bmi_group']].head())
```

```
        age age_group   bmi        bmi_group
id
9046   67.0       65+  36.6  Extremely Obese
51676  61.0     50-65   NaN              NaN
31112  80.0       65+  32.5            Obese
60182  49.0     35-50  34.4            Obese
1665   79.0       65+  24.0           Normal
```

Figure 11.Snapshot of Discretization Method

And we also make clean for our code, so we try to delete a row from our dataset and we display our dataset in several Snapshots by displaying our dataset before and after clean and also during using the method for deleting the row.

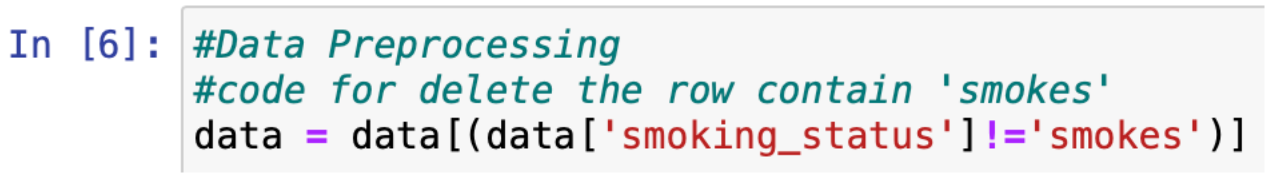


```
In [5]: #Data Preprocessing
        #Stroke prediction dataset before delete the row contain 'smokes'
        data.head()
```

Out[5]:

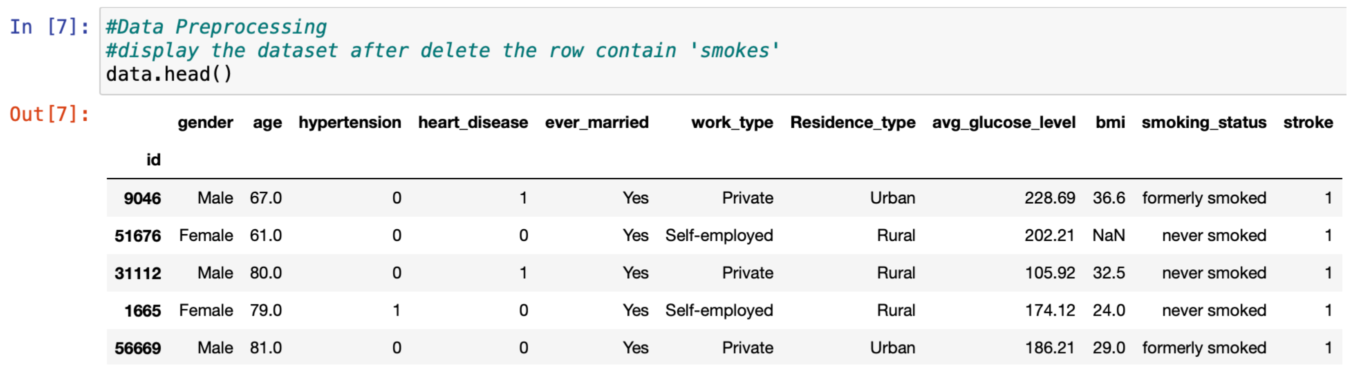
	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
id											
9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
31112	Male	80.0	0	1	Yes	Private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

Figure 12.Snapshot for the dataset before delete



```
In [6]: #Data Preprocessing
        #code for delete the row contain 'smokes'
        data = data[(data['smoking_status']!='smokes')]
```

Figure 13.Snapshot for the dataset during deleting the row



In [7]:
```
#Data Preprocessing
#display the dataset after delete the row contain 'smokes'
data.head()
```

Out[7]:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
id											
9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
56669	Male	81.0	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked	1

Figure 14.Snapshot for the dataset after clean

6. References:

[1] https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/

[2] Ganpathi, A. (2021) *Data preprocessing and Exploratory Data Analysis for machine learning.*, *Medium*. Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/data-preprocessing-and-exploratory-data-analysis-for-machine-learning-75b8a6468b72 (Accessed: April 26, 2023).

[3] Fedesoriano (2021) *Stroke prediction dataset*, *Kaggle*. Available at: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/code?datasetId=1120859&searchQuery=basic (Accessed: April 26, 2023).