IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: Lanka Pathmakumara

Student Reference Number:

Module Code: PUSL2077	Module Name: Data Science in Python
Coursework Title: Final Report	
Deadline Date: 19/03/2024	Member of staff responsible for coursework: Ms. Lakni Peiris
Programme: BSc (Hons) Data Science	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

Lanka Pathmakumara -10899186
Gabbalage Dilshan - 10899287
Ponnahennadige Dias - 10899285
Bathala Wicramasinhe - 10899497
Balasuriya Balasuriya - 10899180

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group:

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed :

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software……………………………………………………………

Overall mark _____% **Assessors Initials ______** **Date__________**

*Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

NSBM
NATIONAL SCHOOL OF BUSINESS MANAGEMENT

IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Data Science in Python

PUSL2077

Statistical Analysis of Stroke-Related Dataset

Group 5

Contents

Introduction

Strokes represent a serious global health risk since they are the primary cause of significant disability and death globally. The need for thorough project and knowledge grows more pressing as the population ages and the prevalence of strokes among young people in low- and middle-income nations rises. To address this need, we conduct a study using Python for statistical analysis on a dataset linked to strokes with the goal of identifying important trends, risk factors, and predicting model connected to strokes. With this project, we hope to lessen the number of stroke-related deaths and impairments by aiding in the creation of efficient treatment and preventative plans.

With 5000 items, our dataset is a good resource for practical knowledge related to stroke-related problems. We aim to uncover important insights through thorough investigation and analysis that will enable health care providers to develop focused interventions for stroke prevention and treatment. By utilizing Python's analytical features, we want to decipher complex linkages and patterns present in the information, ultimately promoting a more profound comprehension of stroke dynamics.

In addition, we document our entire approach in this final report, starting with data loading and exploration and ending with data cleaning, preprocessing, and descriptive analysis. We clarify the procedures used to get the dataset ready for analysis, addressing issues like missing values, outliers, and discrepancies that sprang up during the data preparation stage. We also provide descriptive statistics to reveal underlying data patterns along with data visualization methods to provide a clear understanding of the subtleties of the dataset.

Moreover, we conduct sophisticated statistical analyses, such as regression or correlation analysis, to extract more information about the connections between different dataset attributes. By employing these diverse analytical techniques, we hope to shed light on the complex interactions between the variables that influence the risk of stroke and enable well-informed choices on stroke prevention and management.

The conclusions drawn from our investigation will give medical practitioners important knowledge to help them comprehend the different aspects that lead to strokes and help them create focused prevention and treatment plans. Our work intends to enable the adoption of efficient interventions to lessen the burden of strokes on people and healthcare systems worldwide by identifying important risk factors and their interactions.

The following columns make up the dataset: id, gender, age, heart disease, hypertension, work type, residence type, average blood sugar level, bmi, smoking status, and stroke. Here's a quick rundown of some important characteristics:

- Hypertension: This is the medical term for high blood pressure, which is characterized by a persistently elevated force of blood against the arterial walls. It frequently poses a serious risk for strokes.
- Heart disease: This characteristic shows if a person has a heart condition or not. This is a binary variable, meaning that '0' denotes the absence of cardiac disease and '1' indicates its presence.
- Ever married: This tells if a person has ever been married or not at least once in their life.
- Work_type: Indicates the kind of work that a person is doing. It has sections labeled "Private," "Self-employed," "Govt job," "Children," and "Never worked".
- Residence_type: Denotes whether a person lives in a rural or urban setting.
- Smoking_status: Gives details about a person's smoking preferences. It has options such as "smokes," "never smoked," "formerly smoked," and "Unknown".
- Body Mass Index, or BMI, is a weight-and-height-based indicator of body fat. Weight in kilos divided by height in meters squared is how it is computed.
- The term "Avg_glucose_level" refers to the average blood glucose level, expressed in milligrams per deciliter (mg/dL).

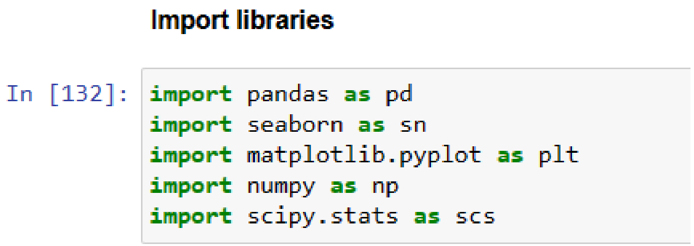
These characteristics will be essential to our investigation in order to comprehend the different stroke-causing elements and to create prediction models for assessing stroke risk.

Data Loading and Exploration

We describe the preliminary work done in our stroke dataset analysis. This include importing the dataset into our analysis platform and performing exploration to learn more about the features, organization, and any trends of the data.

Importing necessary libraries

To make data processing, analysis, and visualization easier, import necessary libraries like pandas, NumPy, matplotlib, and seaborn.

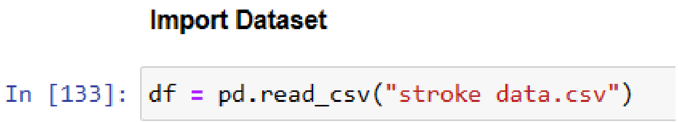


Import libraries

```
In [132]: import pandas as pd
          import seaborn as sn
          import matplotlib.pyplot as plt
          import numpy as np
          import scipy.stats as scs
```

Loading the dataset

The first step was to use the pandas library to load the Stroke dataset into Python. pd.read_csv() was the function we used to read the CSV file into a Data Frame.



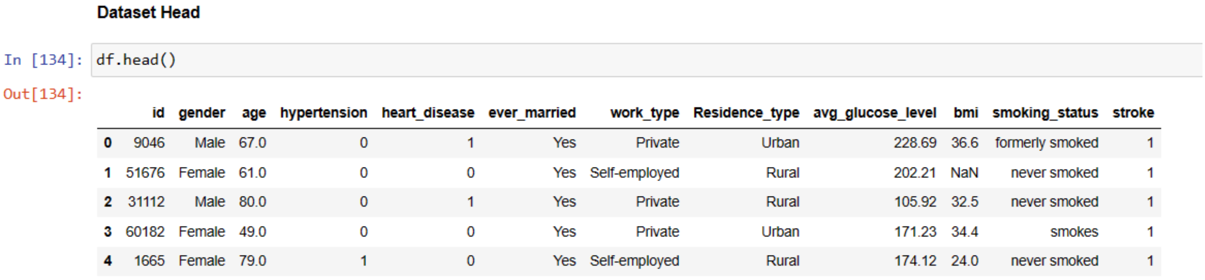
Import Dataset

```
In [133]: df = pd.read_csv("stroke data.csv")
```

After loading the data, we carried out some initial **exploration** tasks like:

Display first few rows of the dataset.

To comprehend the structure of the dataset, head() is used to display the first few rows of the dataset.



Dataset Head

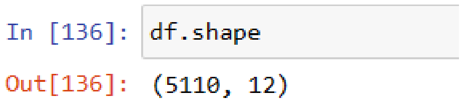
```
In [134]: df.head()
Out[134]:
```

	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

Dimensionality of the dataset.

Utilizing "shape" to determine the dataset's dimensionality.

Dataset shape



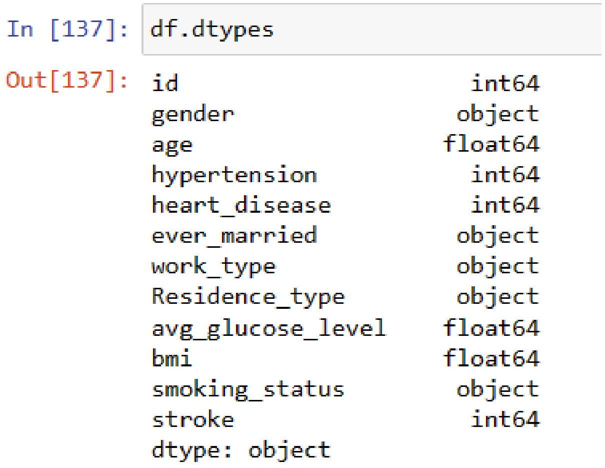
```
In [136]: df.shape
Out[136]: (5110, 12)
```

There are 5110 rows and the 12 columns in this dataset.

Identifying datatypes of the dataset.

Looking at each column's data type to find any discrepancies.

Dataset Data types



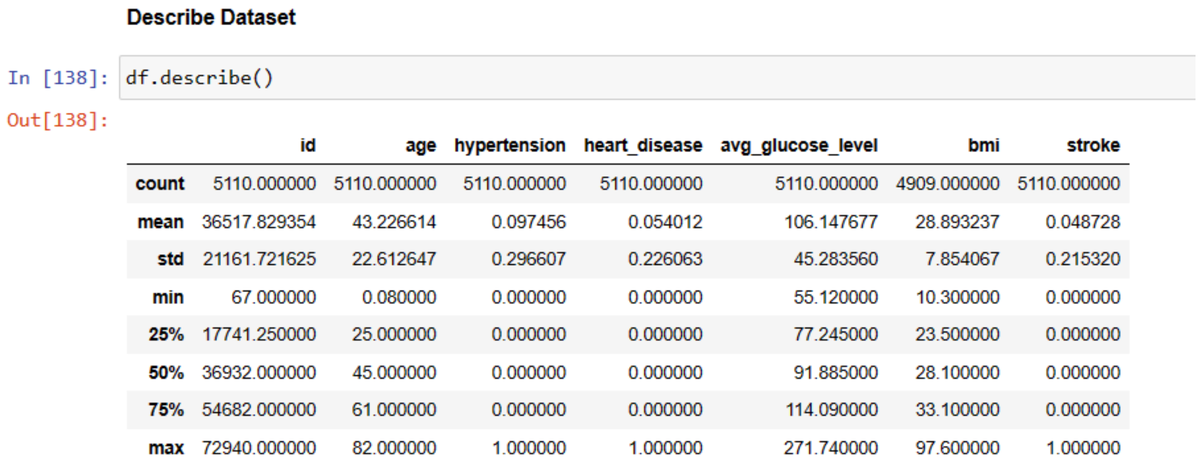
```
In [137]: df.dtypes
Out[137]: id                     int64
          gender                object
          age                  float64
          hypertension           int64
          heart_disease          int64
          ever_married          object
          work_type             object
          Residence_type        object
          avg_glucose_level    float64
          bmi                  float64
          smoking_status        object
          stroke                 int64
          dtype: object
```

Knowing the different sorts of data in a dataset is essential for:

➢ appropriate data analysis and modification.
efficiency of memory.
ensuring precise calculations and processes.
assisting with preprocessing and data cleaning.
Improving model efficiency in activities using machine learning.

Summarizing basic statistics

Utilizing "describe()" to summarize fundamental statistics and obtain understanding of the numerical variables.



Describe Dataset

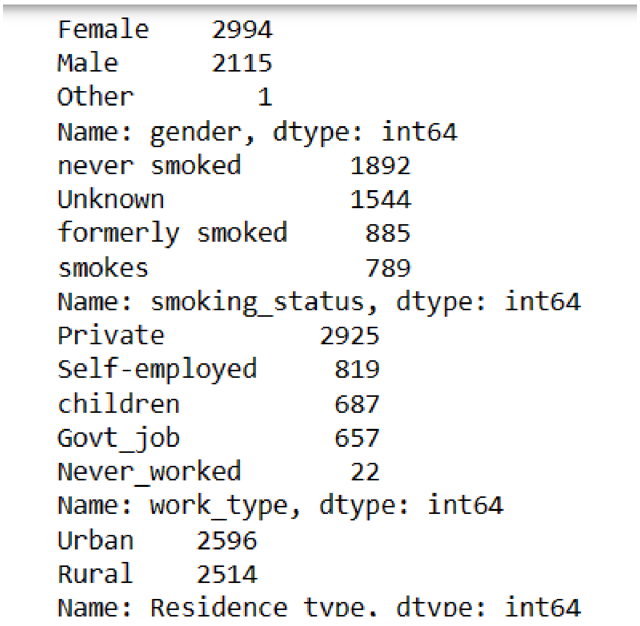
```
In [138]: df.describe()
Out[138]:
```

	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

Key summary statistics of a DataFrame, such as count, mean, standard deviation, minimum, and maximum values, are provided by the df.describe() function, providing a brief synopsis of the distribution and range of the data.

Check unique values and their frequencies for categorical variables.

Next we get an idea about the unique values and their frequencies. We considered gender, smoking_status, work_type, Residence_type columns. Those columns are categorical columns.



```
Female     2994
Male       2115
Other         1
Name: gender, dtype: int64
never smoked       1892
Unknown            1544
formerly smoked     885
smokes              789
Name: smoking_status, dtype: int64
Private          2925
Self-employed     819
children          687
Govt_job          657
Never_worked       22
Name: work_type, dtype: int64
Urban     2596
Rural     2514
Name: Residence_type, dtype: int64
```

In the gender column we identified 2994-females,2115-males and 1-other. In smoking_status column we found 1892-never smoked, 1544-unknown, 885-formerly smoked and 789-smokes. In work_type column, there are 2925-private , 819-self employed,687-children, 657-govt_job and 22-never worked. In residence type , 2596-urban and 2514-rural.

These results give us information on how categorical variables are distributed throughout the dataset, which aids in our comprehension of the diversity and frequency of each category. For additional dataset analysis and interpretation, this information is essential.

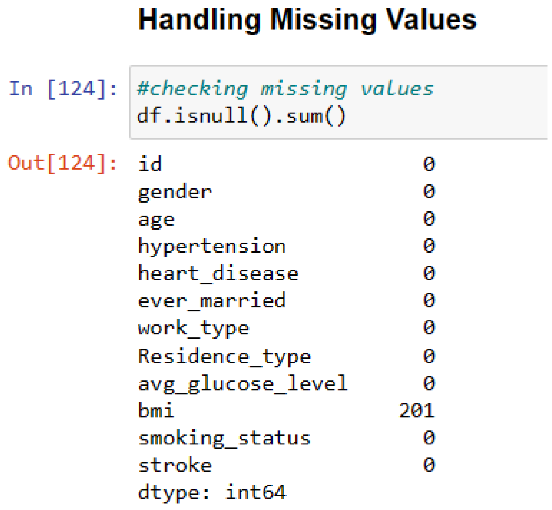
Challenges encountered and Solutions.

- Missing Values: employing suitable statistical techniques for their imputation.
- Outliers: Identified and managed to guarantee reliable analytical findings utilizing statistical methods like Z-score or IQR.

Data Cleaning and Preprocessing

Handling missing values

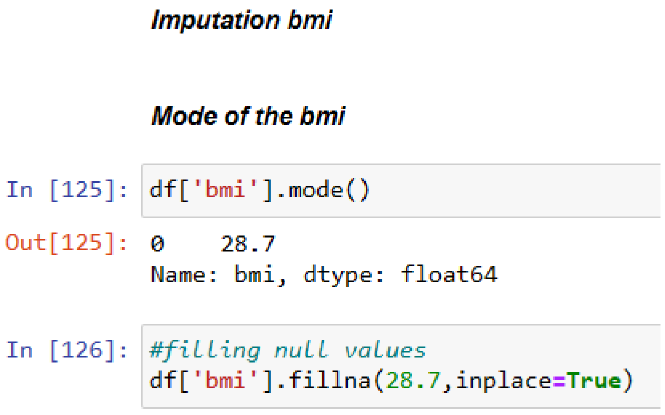
First, we identified missing values in the dataset. We only found missing values in the bmi column.



Handling Missing Values

```
In [124]: #checking missing values
          df.isnull().sum()

Out[124]: 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
```

There are various methos to handle missing values. Such as imputation, deletion, prediction etc. We used imputation to handle missing values. There are three methods in imputation. Those are, Mean/Median/Mode imputation: Use the column's mean, median, or mode to fill in any missing values. We used mode imputation.



Imputation bmi

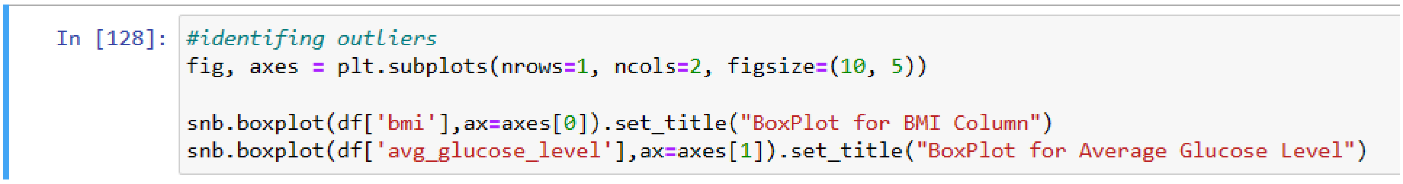
Mode of the bmi

```
In [125]: df['bmi'].mode()

Out[125]: 0    28.7
          Name: bmi, dtype: float64

In [126]: #filling null values
          df['bmi'].fillna(28.7,inplace=True)
```

Handling Outliers

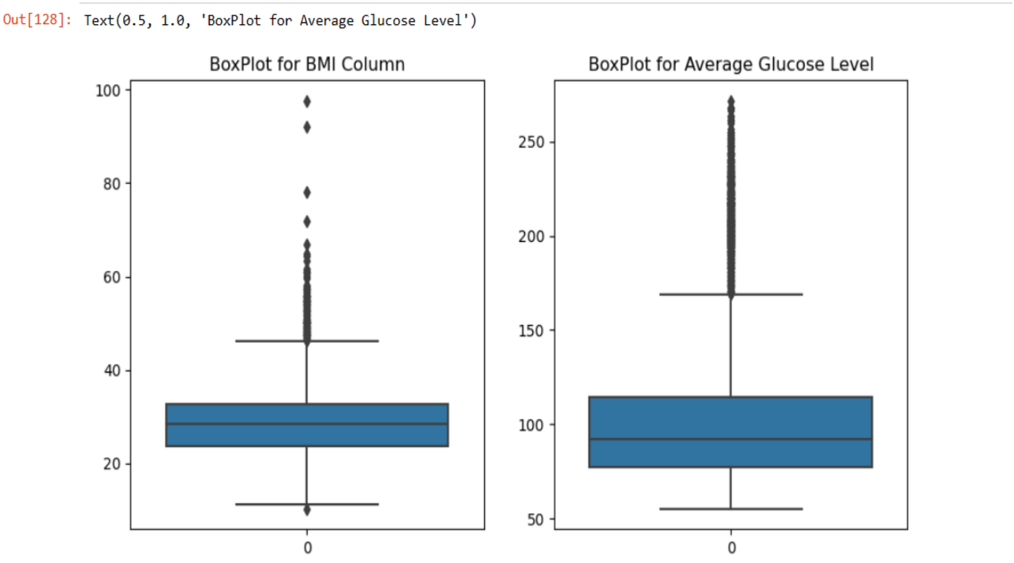
The accuracy and dependability of statistical studies and machine learning models can be considerably impacted by outliers. Therefore, removing outliers is very important. There are some techniques to identify outliers.

They are Z-score method, inter quartile method, visualization method etc. So, we used the visualization method to identify outliers. We used boxplots to detect outliers.



```
In [128]: #identifing outliers
          fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))

          snb.boxplot(df['bmi'],ax=axes[0]).set_title("BoxPlot for BMI Column")
          snb.boxplot(df['avg_glucose_level'],ax=axes[1]).set_title("BoxPlot for Average Glucose Level")
```

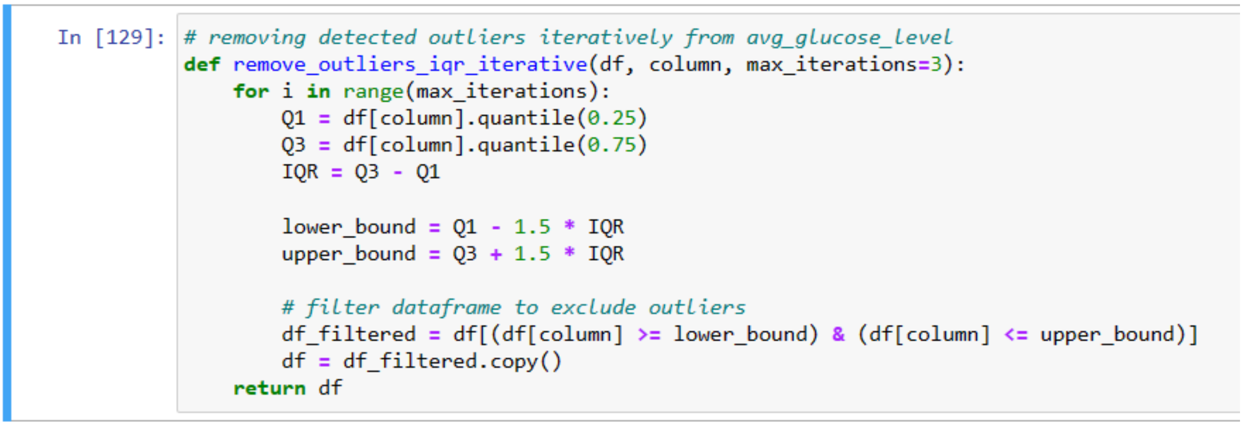
We found some outliers from the bmi column and average glucose level column.



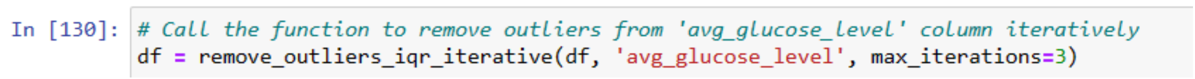
```
Out[128]: Text(0.5, 1.0, 'BoxPlot for Average Glucose Level')
```

We removed all the identified outliers as the next step. We used an iterative outlier removing method. Iterative outlier removal is a methodical process that repeatedly finds and eliminates extreme data points from a dataset. This technique progressively refines the dataset by repeating the outlier elimination process, lessening the impact of outliers and enhancing the general dependability of the data for analysis.

We removed outliers from the average glucose level column.

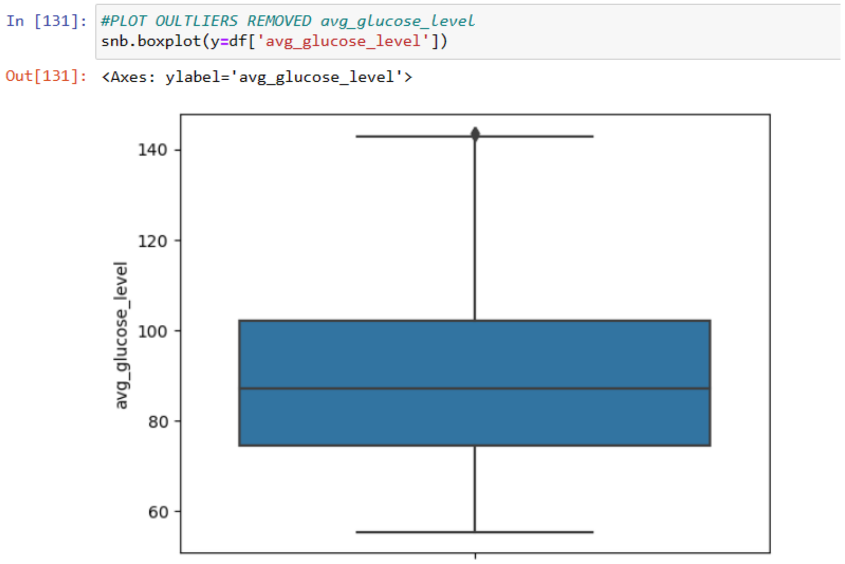


```
In [129]: # removing detected outliers iteratively from avg_glucose_level
          def remove_outliers_iqr_iterative(df, column, max_iterations=3):
              for i in range(max_iterations):
                  Q1 = df[column].quantile(0.25)
                  Q3 = df[column].quantile(0.75)
                  IQR = Q3 - Q1

                  lower_bound = Q1 - 1.5 * IQR
                  upper_bound = Q3 + 1.5 * IQR

                  # filter dataframe to exclude outliers
                  df_filtered = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]
                  df = df_filtered.copy()
              return df
```



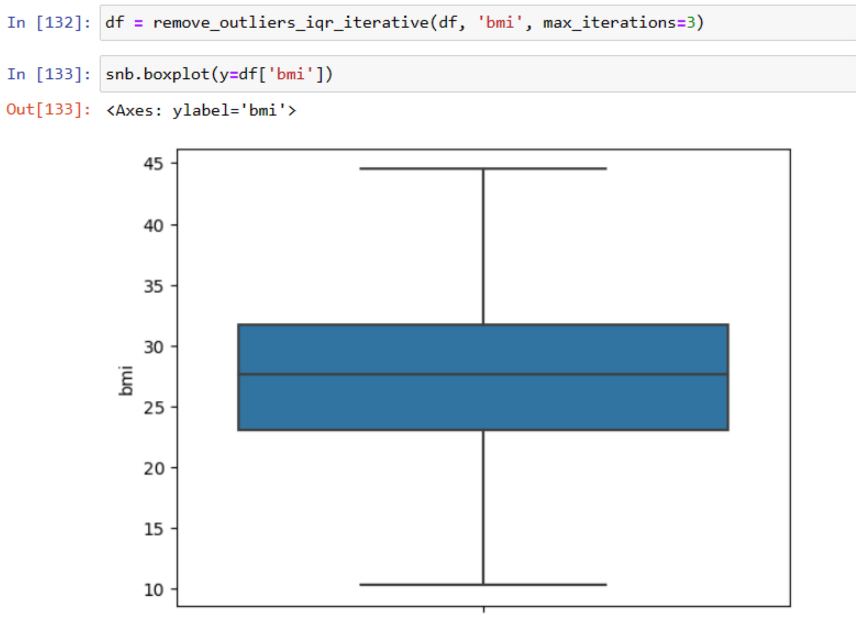
```
In [130]: # Call the function to remove outliers from 'avg_glucose_level' column iteratively
          df = remove_outliers_iqr_iterative(df, 'avg_glucose_level', max_iterations=3)
```

After removing outliers Average glucose level boxplot shown below.



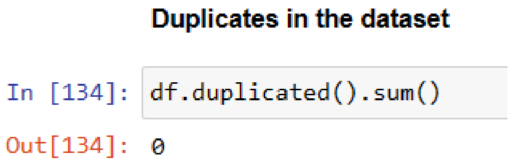
```
In [131]: #PLOT OULTLIERS REMOVED avg_glucose_level
          snb.boxplot(y=df['avg_glucose_level'])

Out[131]: <Axes: ylabel='avg_glucose_level'>
```

At last we removed outliers form the bmi column using the previous iterative method.



```
In [132]: df = remove_outliers_iqr_iterative(df, 'bmi', max_iterations=3)

In [133]: snb.boxplot(y=df['bmi'])

Out[133]: <Axes: ylabel='bmi'>
```

Identifying duplicate values.



Duplicates in the dataset

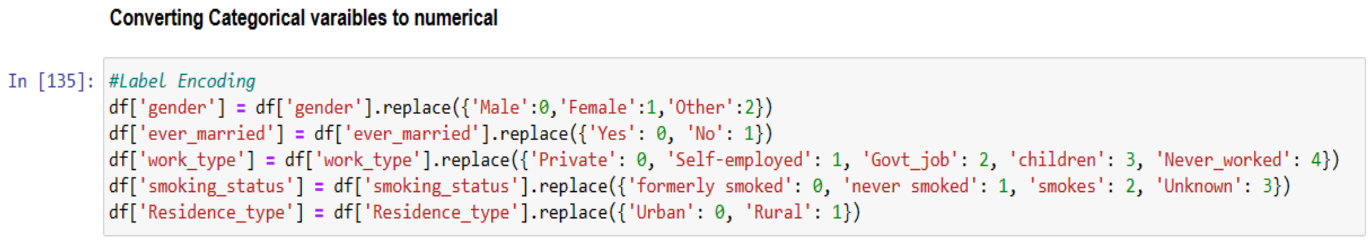
```
In [134]: df.duplicated().sum()

Out[134]: 0
```

Next we need to identify duplicate values. In order to guarantee data integrity and correctness in analysis, it is essential to check a dataset for duplicate values. By distorting statistical measurements like mean and standard deviation, duplicate values can have a substantial impact on analysis. This might result in biased findings and erroneous insights. Duplicates can also skew visualization results, making it more difficult to understand data trends and patterns. Analysts can alleviate these problems and guarantee that analytic outputs are trustworthy, objective, and representative of the actual underlying data distribution by locating and eliminating duplicate values. This procedure is necessary to preserve the quality of the data and optimize the performance of the analytical methods used on the dataset.

There's no any missing values in the dataset.

Encoding

Since many machine learning methods require numerical input, encoding is an essential step in the preprocessing of data, especially when working with categorical variables. Encoding ensures interoperability with these methods by converting categorical data into a numerical representation. Label encoding, one-hot encoding, and ordinal encoding are a few examples of the different encoding techniques. We used Label encoding. Label encoding is appropriate for ordinal categorical data where order counts since it gives each category a distinct numerical label.



Converting Categorical varaibles to numerical

```
In [135]: #Label Encoding
          df['gender'] = df['gender'].replace({'Male':0,'Female':1,'Other':2})
          df['ever_married'] = df['ever_married'].replace({'Yes': 0, 'No': 1})
          df['work_type'] = df['work_type'].replace({'Private': 0, 'Self-employed': 1, 'Govt_job': 2, 'children': 3, 'Never_worked': 4})
          df['smoking_status'] = df['smoking_status'].replace({'formerly smoked': 0, 'never smoked': 1, 'smokes': 2, 'Unknown': 3})
          df['Residence_type'] = df['Residence_type'].replace({'Urban': 0, 'Rural': 1})
```

Descriptive Analysis

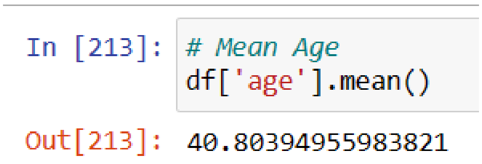
We present summary statistics that include measures of variability and central tendency in the section on descriptive analysis. These statistics provide a thorough overview of the dataset, highlighting important patterns, distributions, and data point dispersion. Our goal in conducting this inquiry is to identify fundamental patterns present in the dataset, which will serve as a basis for further investigation and analysis.

Central Tendency

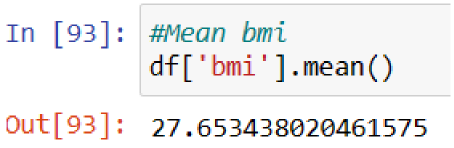
The typical or core value that data points tend to cluster around can be inferred from central tendency measurements like mean, median, and mode. These statistics help you understand the overall distribution and properties of the dataset by providing a concise overview of its central tendency.

Mean

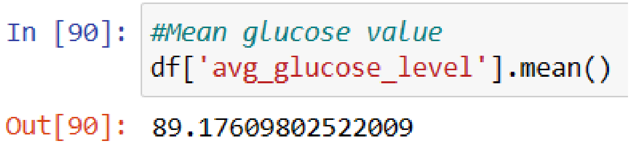
We considered some important columns to get mean values. We used age , avg_glucose_level , bmi columns to get the mean values.



```
In [213]: # Mean Age
          df['age'].mean()

Out[213]: 40.80394955983821
```



```
In [93]: #Mean bmi
         df['bmi'].mean()

Out[93]: 27.653438020461575
```



```
In [90]: #Mean glucose value
         df['avg_glucose_level'].mean()

Out[90]: 89.17609802522009
```

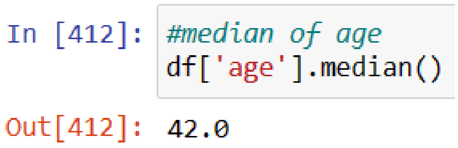
The mean values we computed for our central tendency analysis were as follows:

- It was 40.80 years old on average.
- The glucose level was 89.18 mg/dL on average.
- Body Mass Index (BMI) was 27.65 on average.

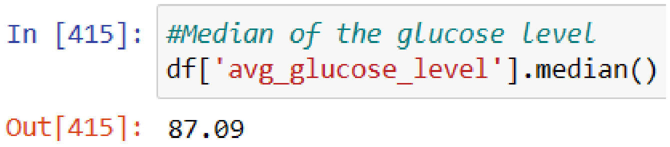
These numbers revealed information on the dataset's typical or average age, BMI, and glucose level. All things considered, these mean values provide an overview of the age, glucose level, and BMI central patterns in the dataset, giving important information on the health traits of the population under study.

Median

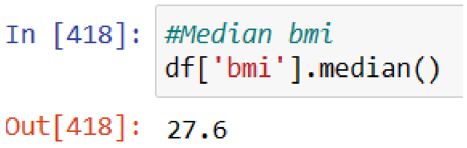
We considered age , avg_glucose_level , bmi columns to get the median values.



```
In [412]: #median of age
          df['age'].median()

Out[412]: 42.0
```



```
In [415]: #Median of the glucose level
          df['avg_glucose_level'].median()

Out[415]: 87.09
```

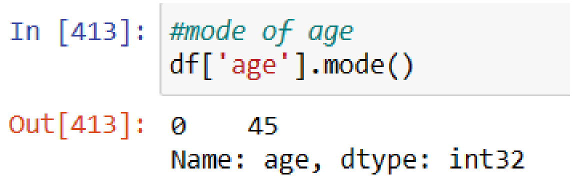


```
In [418]: #Median bmi
          df['bmi'].median()

Out[418]: 27.6
```

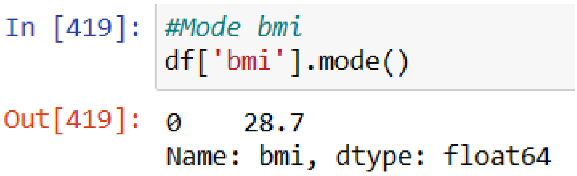
The center value of a dataset when sorted in ascending order is represented by the median, a measure of central tendency. The median is a resilient metric, especially for skewed distributions, since it is not impacted by outliers, in contrast to the mean, which is affected by extreme values. In this particular context, the median values for the variables of age, glucose level, and body mass index (BMI) were determined to be 42.0 years, 87.09 mg/dL, and 27.6 kg/m², respectively. The central tendencies within each variable are helpfully summarized by these median values, which also reveal the usual or central value that the data tends to cluster around.

Mode

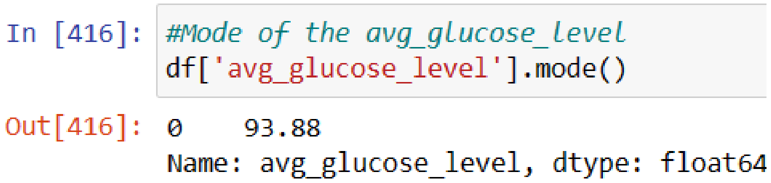
We used several columns to find mode values. Age , average glucoce level, BMI, Hypertension, Ever married , heart disease ,work type , smoking status,Ressidence type columns we used to identify mode values.



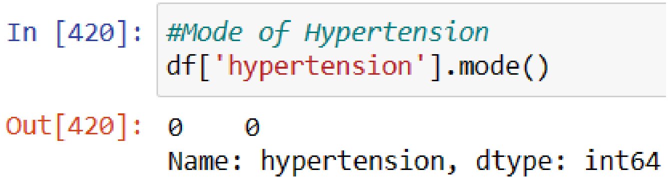
```
In [413]: #mode of age
          df['age'].mode()

Out[413]: 0    45
          Name: age, dtype: int32
```



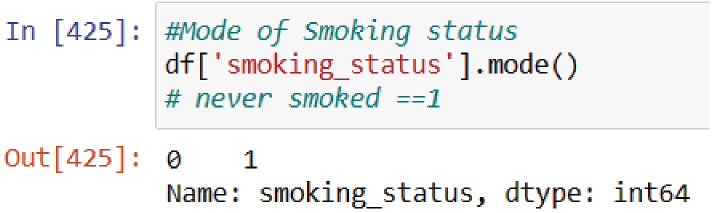
```
In [419]: #Mode bmi
          df['bmi'].mode()

Out[419]: 0    28.7
          Name: bmi, dtype: float64
```



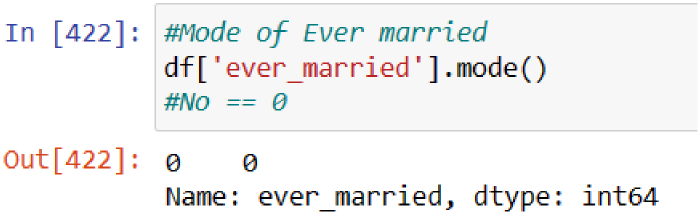
```
In [416]: #Mode of the avg_glucose_level
          df['avg_glucose_level'].mode()

Out[416]: 0    93.88
          Name: avg_glucose_level, dtype: float64
```



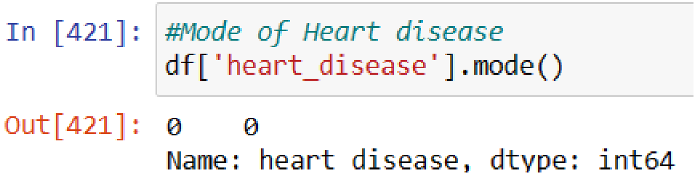
```
In [420]: #Mode of Hypertension
          df['hypertension'].mode()

Out[420]: 0    0
          Name: hypertension, dtype: int64
```



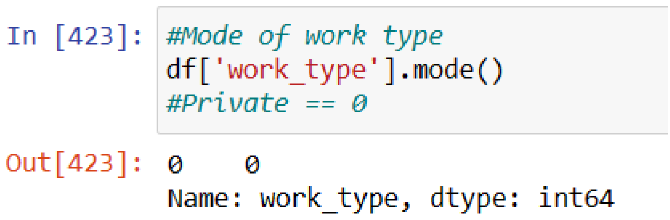
```
In [425]: #Mode of Smoking status
          df['smoking_status'].mode()
          # never smoked ==1

Out[425]: 0    1
          Name: smoking_status, dtype: int64
```



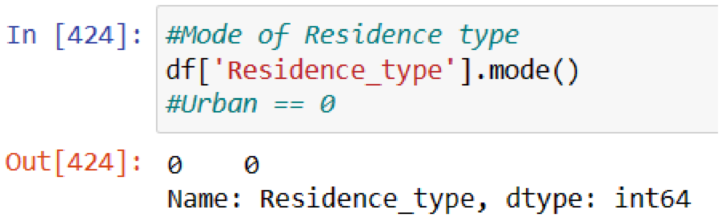
```
In [422]: #Mode of Ever married
          df['ever_married'].mode()
          #No == 0

Out[422]: 0    0
          Name: ever_married, dtype: int64
```



```
In [421]: #Mode of Heart disease
          df['heart_disease'].mode()

Out[421]: 0    0
          Name: heart disease, dtype: int64
```



```
In [423]: #Mode of work type
          df['work_type'].mode()
          #Private == 0

Out[423]: 0    0
          Name: work_type, dtype: int64
```



```
In [424]: #Mode of Residence type
          df['Residence_type'].mode()
          #Urban == 0

Out[424]: 0    0
          Name: Residence_type, dtype: int64
```

In our analysis of the mode for various variables:

- Age: The dataset's most often observed age is 45, as indicated by the mode age of the data.
- BMI: 28.7 is the modal BMI, indicating that this is the most common BMI value.
- Average Blood Sugar Level: With a mode average of 93.88, this is the most frequently observed blood sugar level.
- Hypertension: Based on the hypertension mode, it appears that the majority of the dataset's participants do not have hypertension.
- Smoking Status: Based on the dataset's mode, the majority of participants had never smoked.
- Ever Married: The majority of individuals in the dataset are single, according to the mode for the ever married column.
- Heart Disease: The heart disease column's mode indicates that the majority of the dataset's participants do not have heart disease.
- job Type: Based on the mode for job type, the majority of the dataset's participants are employed privately.
- Residence Type: Based on the dataset's mode, the majority of its members live in cities.

These mode values give an overview of the features and distributions of the dataset by illuminating the most frequent or typical observations within each corresponding variable.

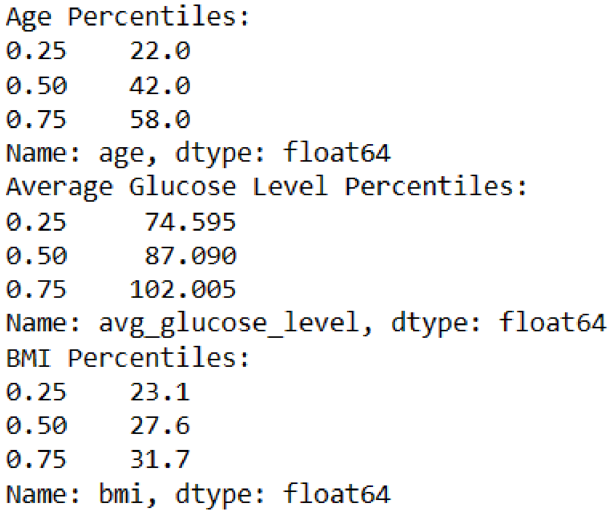
All things considered, examining mode values offers a glimpse of the most common or usual observations within each variable, providing important information about the features of the dataset and any patterns. These revelations can direct additional research and decision-making procedures across a range of industries, including the medical field.

Measures of Position

Position measures, especially percentiles, are statistical tools that are used to partition a dataset into equal segments according to their relative positions. Percentiles are useful tools for analyzing data distribution since they show where specific observations lie in the dataset in relation to other observations. The 50th percentile, for instance, is represented by the median and denotes the number below which half of the data points are located.

When analyzing the distribution of data and spotting possible outliers or extreme results, percentiles are quite helpful. Analysts can evaluate the dataset's central tendency and variability and decide on data handling and analysis tactics by computing percentiles.

We found percentiles as follows,



```
Age Percentiles:
0.25    22.0
0.50    42.0
0.75    58.0
Name: age, dtype: float64
Average Glucose Level Percentiles:
0.25     74.595
0.50     87.090
0.75    102.005
Name: avg_glucose_level, dtype: float64
BMI Percentiles:
0.25    23.1
0.50    27.6
0.75    31.7
Name: bmi, dtype: float64
```

Regarding the distribution of ages:

- According to the 25th percentile (Q1), over 25% of the dataset's members are younger than or equal to 0.22.
- About 0.50 is the median (50th percentile, Q2) age, meaning that half of the population is younger than or equal to this age 42.
- The 75th percentile (Q3), which is roughly 0.75, indicates that 75% of people are either younger than or equal to this age 58.
- As a result, you would correctly characterize the distribution as negatively skewed, with the bulk of people being younger than the median age.

Considering the distribution of average glucose levels:

- According to the 25th percentile (Q1), around 25% of people have blood glucose levels that are less than or equal to 74.595.
- Half of the people have glucose levels that are lower than or equal to the median (50th percentile, Q2), which is approximately 87.090.
- The 75th percentile (Q3), which is roughly 102.005, indicates that 75% of people have glucose levels that are lower than or equal to this amount.
- Because the median is closer to Q1, it is assumed that the distribution is negatively skewed.
-

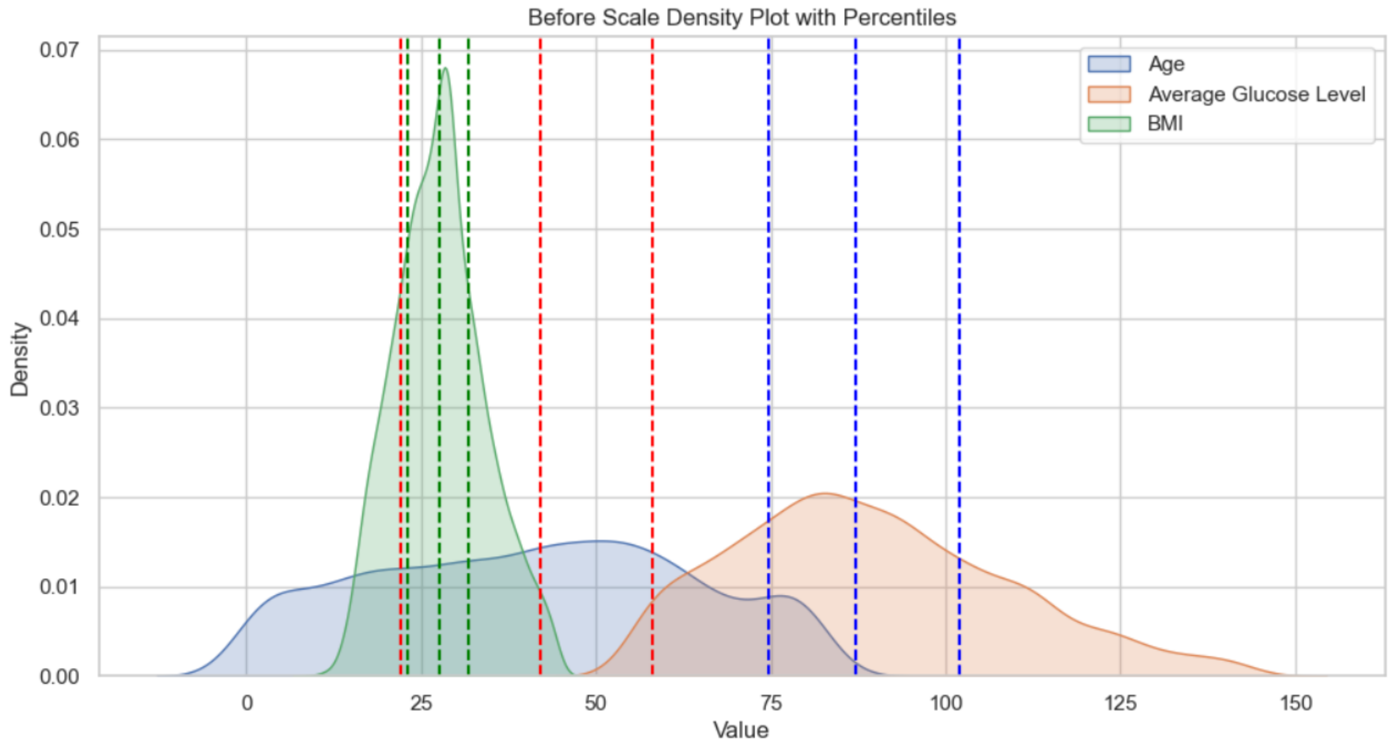
Regarding the distribution of BMI:

- According to the 25th percentile (Q1), about 25% of people have a BMI of less than or equal to 23.1.
- Half of the population is thought to have a BMI that is less than or equal to the median (50th percentile, Q2) BMI of roughly 27.6.
- About 75% of people have a BMI that is less than or equal to the 75th percentile (Q3), which is roughly 31.7.
- Because the median is closer to Q1, it is assumed that the distribution is negatively skewed.

Gaining knowledge of these variables' distributional properties, such as skewness and central tendency, can help understand how they relate to the target variable. For example, in predictive modeling tasks, measures of central tendency such as the median might be crucial predictors or features in the model, but skewed distributions may need to be transformed to improve model performance. Furthermore, by comprehending the distribution of these variables, one can guide future analyses or interventions by identifying potential risk factors or predictors for the objective variable (such as stroke).

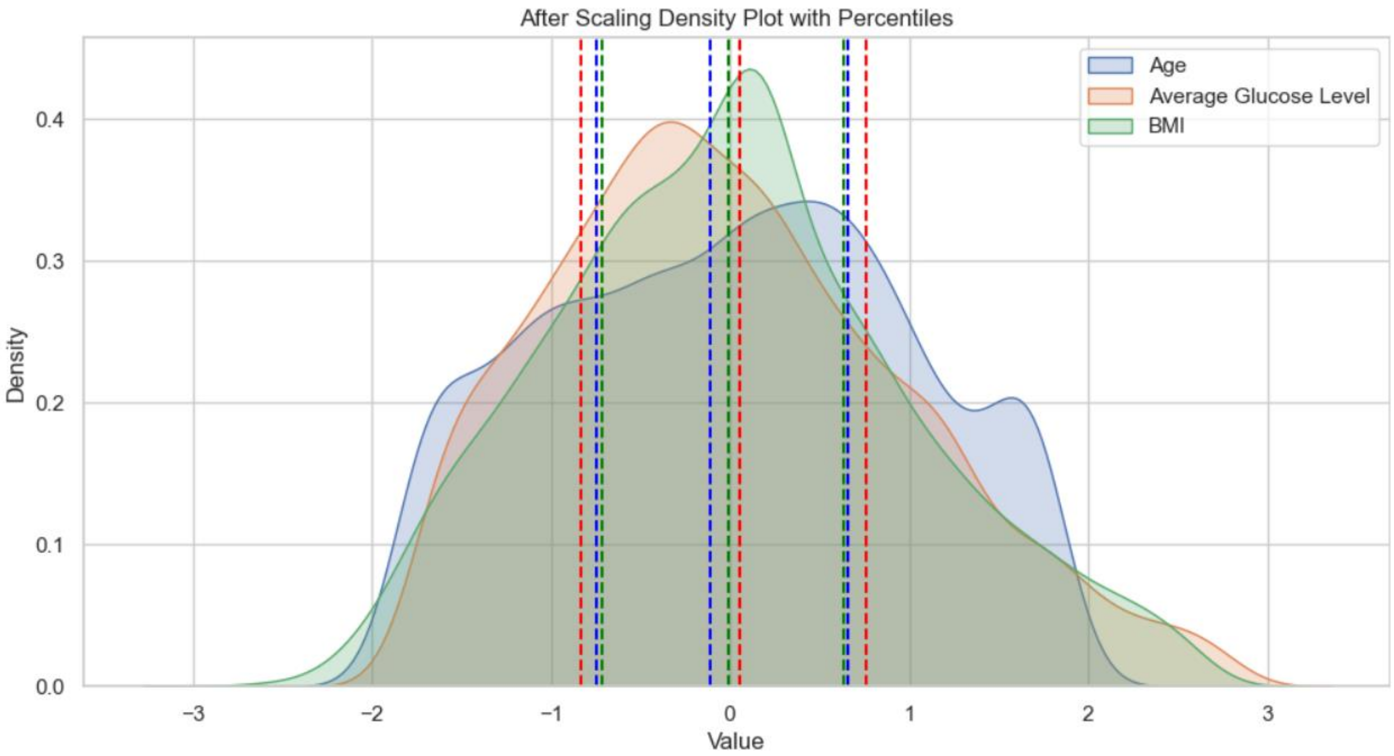
Spread of Columns.

The spread and central tendencies of the age, average glucose level, and BMI variables are better understood thanks to this visualization, which shows the distributions of these variables along with their percentiles.



Feature Scaling

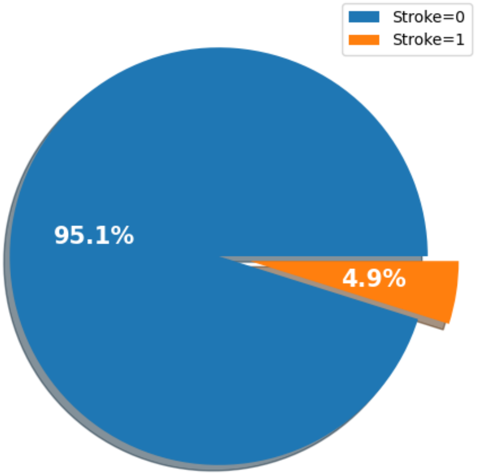
Next, we use StandardScaler from the sklearn.preprocessing module to conduct feature scaling on the numerical columns ('age', 'avg_glucose_level', and 'bmi'). By standardizing the numerical values to have a mean of 0 and a standard deviation of 1, this scaling procedure avoids any problems arising from the different scales of the variables and makes the results comparable.



To guarantee that every feature contributes equally to the analysis and produce a more accurate model performance, "feature scaling" is crucial. Understanding the transformation that the variables go through and how it affects the analysis is made easier by visualizing the scaled distributions using percentiles. This, in turn, leads to a deeper comprehension of the dataset and its properties.

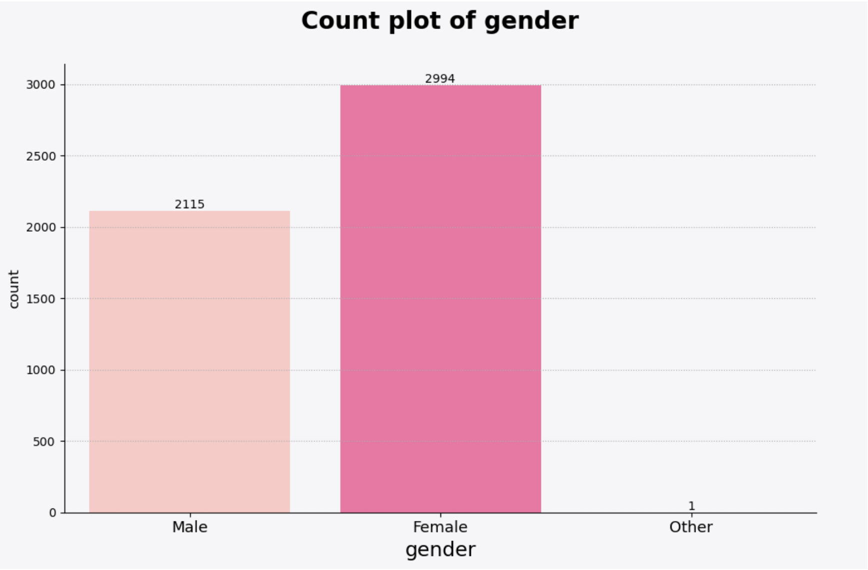
Data Visualization

Stroke patients.



To better comprehend the target variable's prevalence within the dataset, we used a pie chart to show its distribution. 95.1% of the cases had no strokes, according to the research, whereas 4.9% of the cases did. We were given a baseline understanding of the rate of stroke occurrence in the dataset via this depiction.

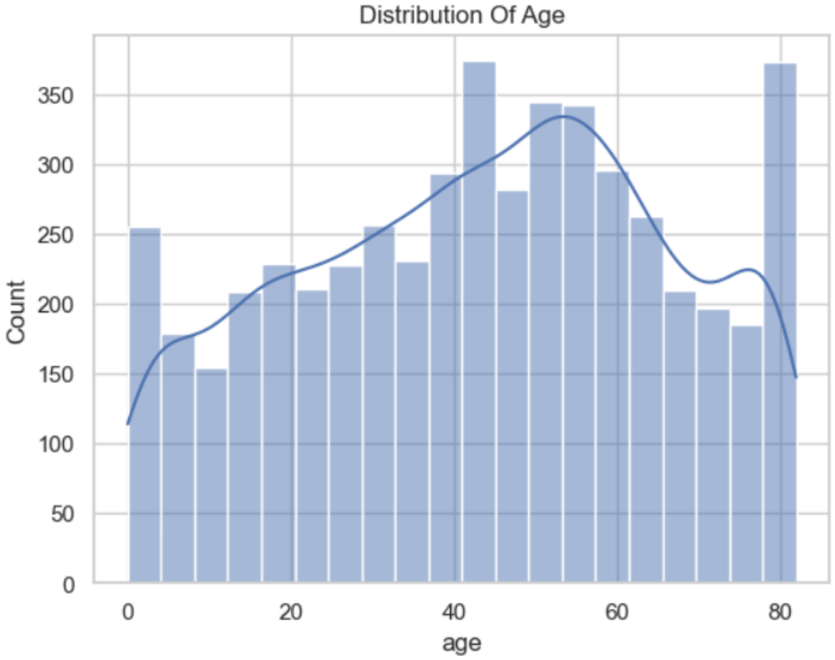
Count plot for gender.



Using a count plot, we were able to identify 2115 males and 2994 females. We were able to compare the gender distribution within the dataset thanks to this visualization, which gave us information on the relative proportions of men and women.

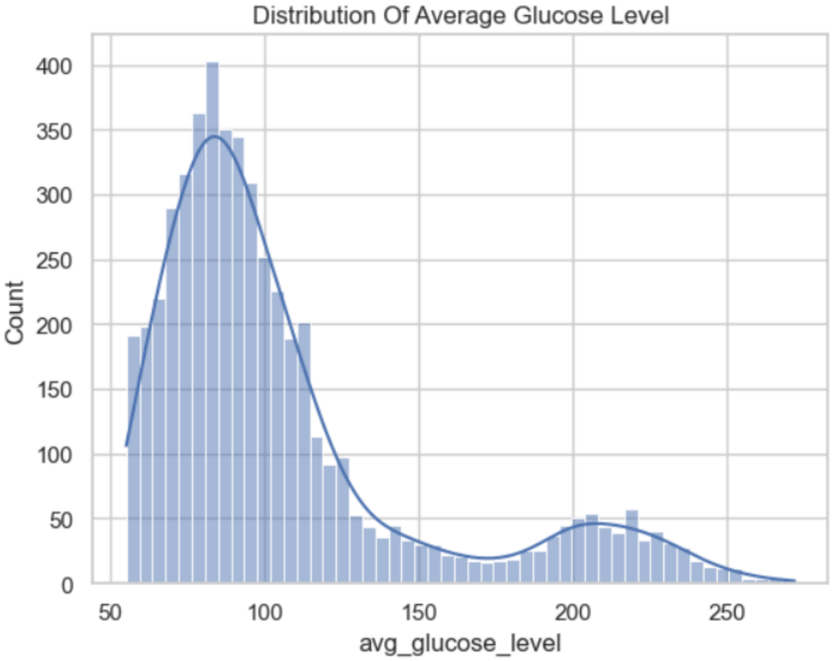
Age Distribution

We may evaluate the age distribution's correlation with the target variable by looking at it. Age may be a significant factor determining the occurrence of the target condition, for example, if there are discernible disparities in the age distribution between persons with and without the condition (e.g., stroke). Furthermore, determining whether the age distribution is skew or multimodal might shed light on the properties of the dataset and direct future research.



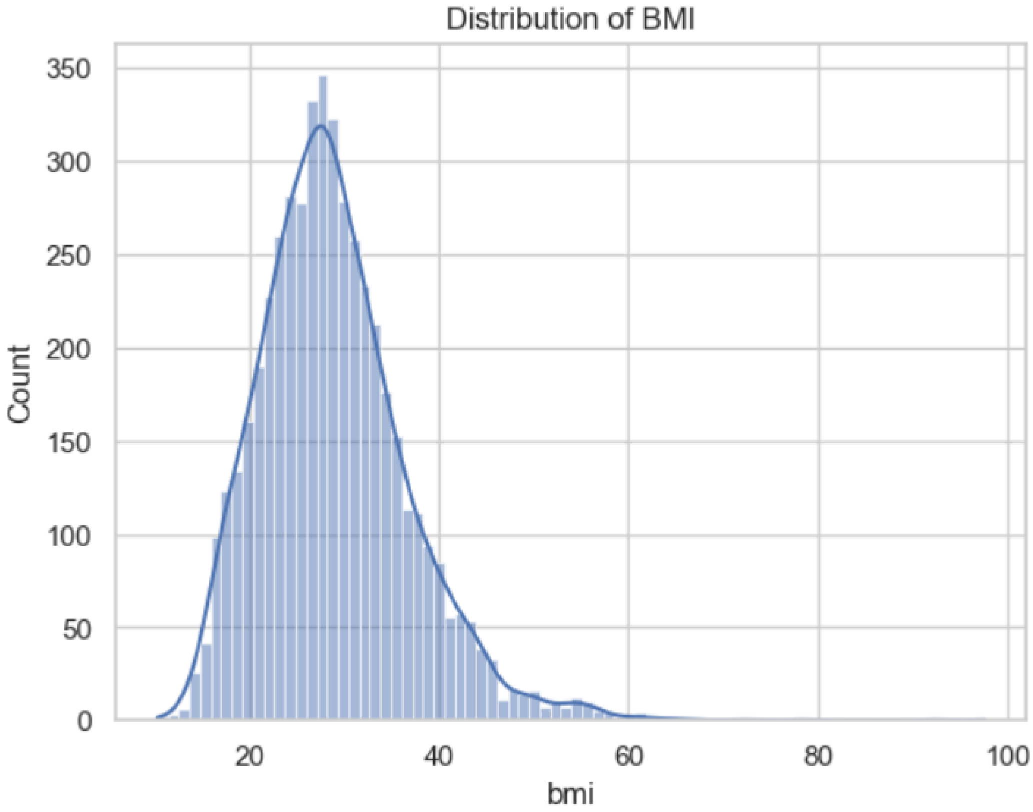
The age histogram plot helps identify significant predictors or risk factors linked to the goal condition by allowing us to examine the distribution of age and assess its possible influence on the target variable.

Average glucose levels distribution



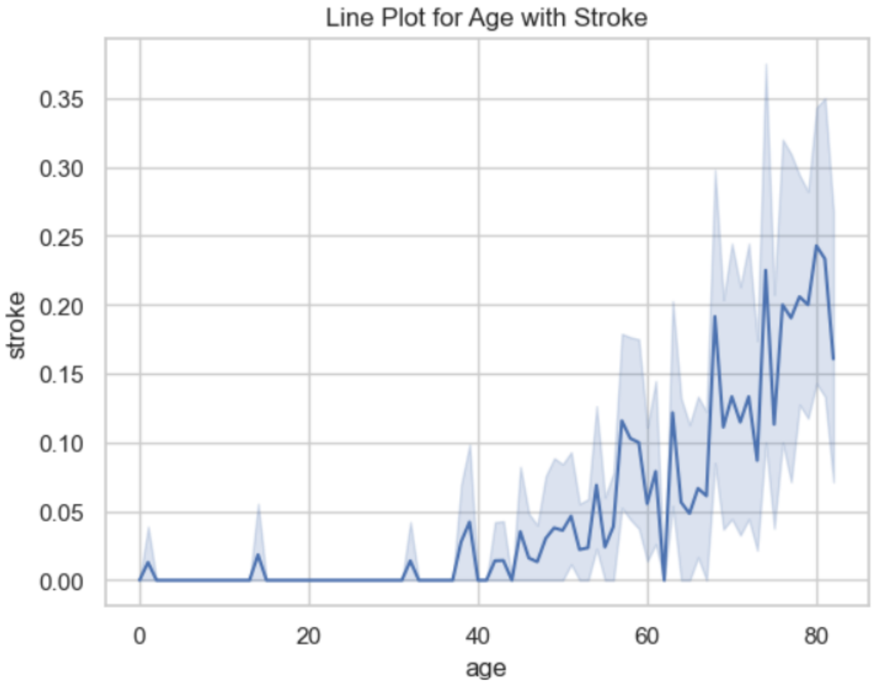
We may investigate the distribution of average glucose levels and assess their possible influence on the goal variable using the histogram plot, which helps identify significant risk factors or predictors linked to the target condition.

Distribution of BMI



In order to help identify significant predictors or risk factors connected to the target condition, the BMI histogram plot enables us to examine the distribution of the variable and assess its possible influence on the target variable.

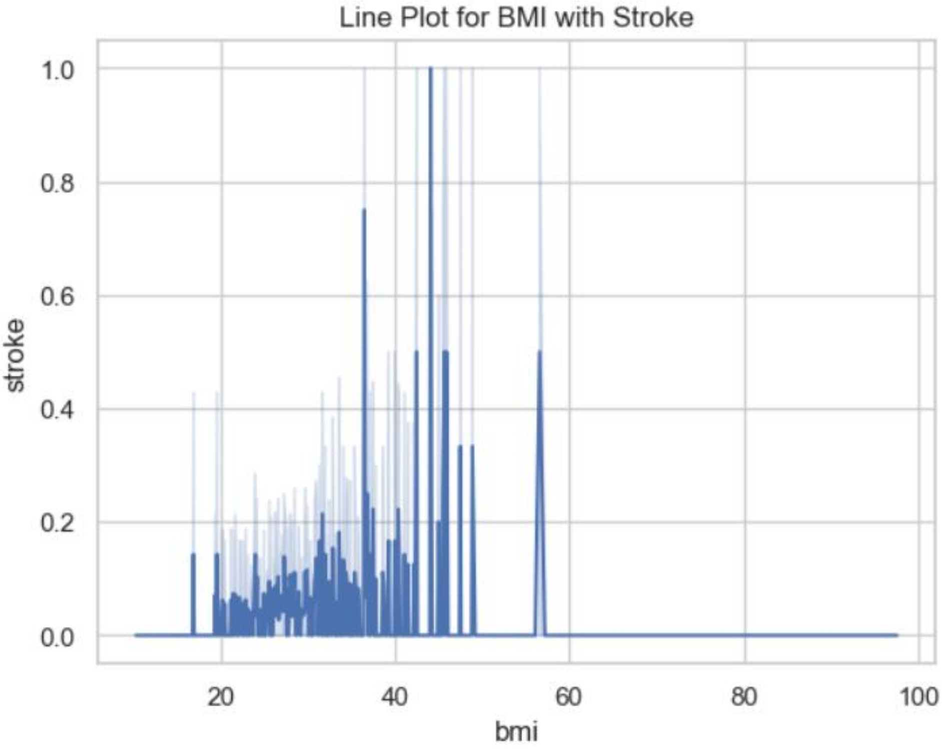
Line Plot age with stroke.



We can evaluate how the incidence of strokes changes with age by looking at the line plot. Notable oscillations or patterns in the line can point to age-related trends in the incidence of stroke. For example, if the line shows an increasing tendency as age increases, it indicates that strokes are more common in older people.

To summarise, the age line plot associated with strokes facilitates the investigation of the correlation between age and stroke incidence, offering valuable perspectives on the possible influence of age as a risk factor or predictor for strokes within the dataset.

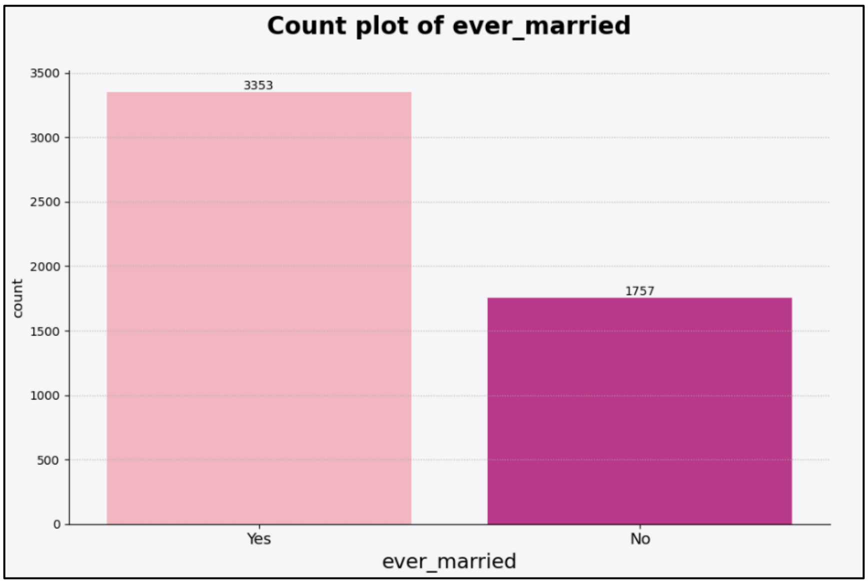
Line plot BMI with stroke.



We may evaluate how the incidence of strokes fluctuates with BMI by looking at the line plot. Notable variations or patterns in the line can point to trends in the incidence of stroke that are connected to BMI. For example, if the line shows an upward trend as BMI increases, this indicates that those with higher BMI values have a higher risk of stroke.

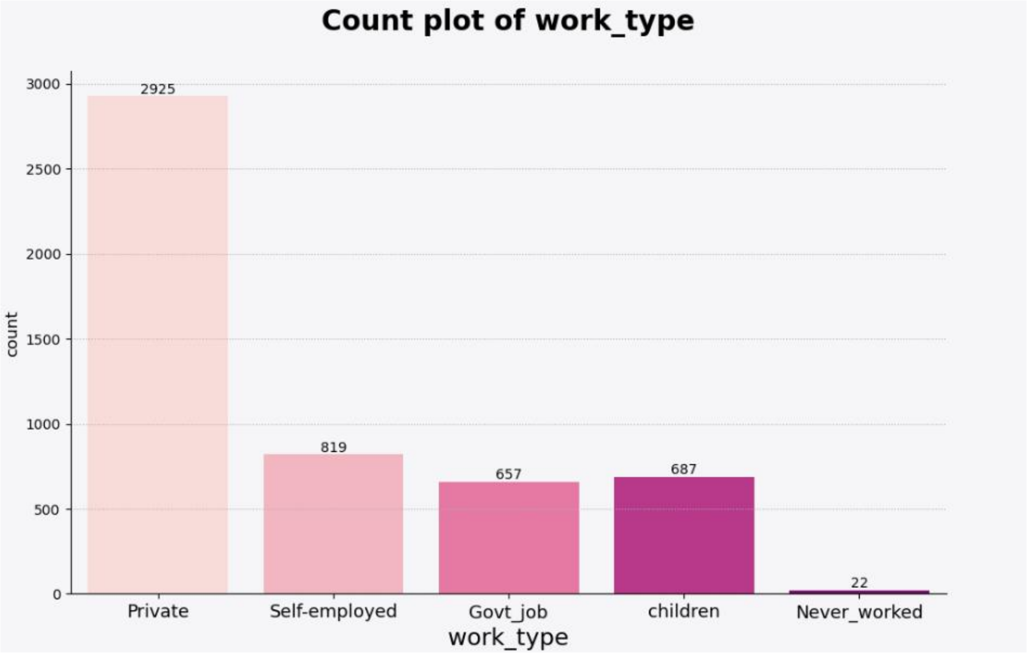
In conclusion, the line plot of BMI with strokes enables us to investigate the connection between BMI and the incidence of strokes, offering information on the possible influence of BMI as a risk factor or predictor for strokes within the dataset.

Count plot for marital status.



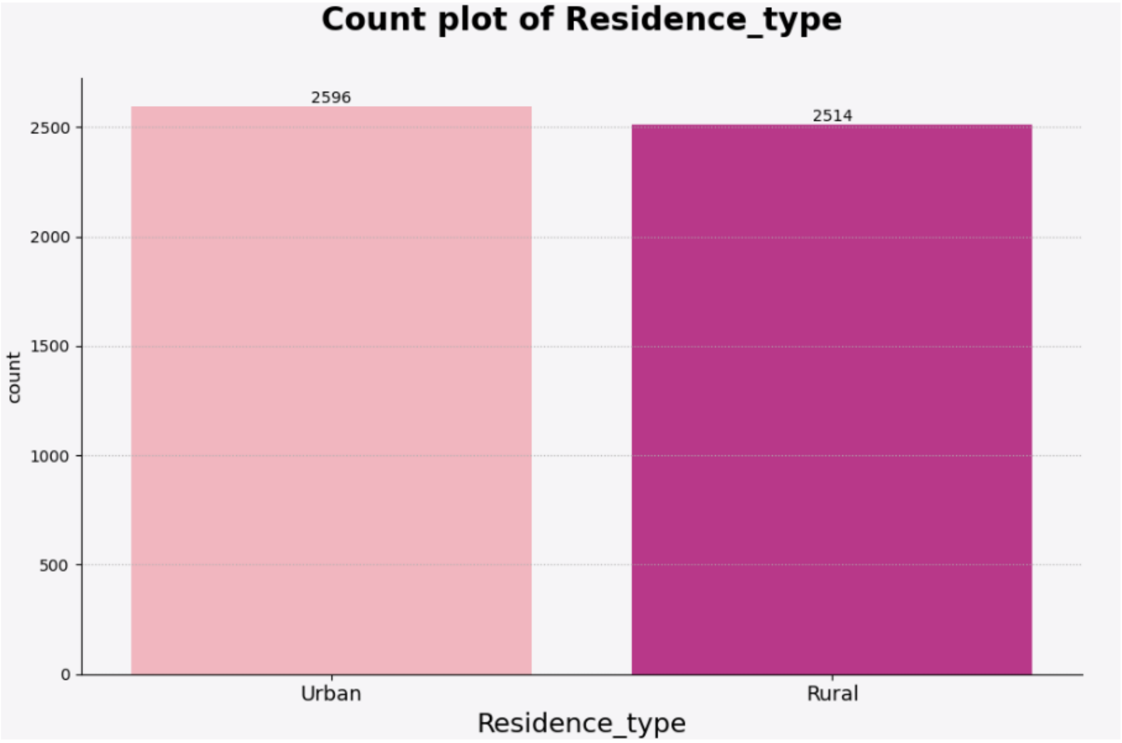
It was determined that 3353 people were married and 1757 people were single. We utilized a count plot to visually compare the number of married and single individuals in the dataset as our representation of this information.

Count plot for work type



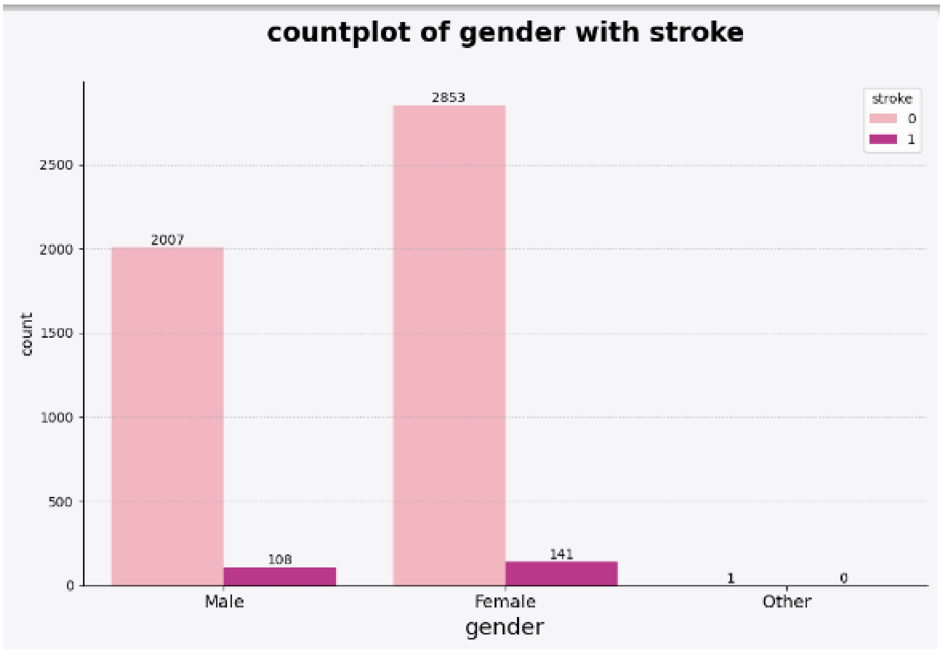
The following counts were shown when we plotted the distribution of the work type column: children - 687, never worked - 22, private - 2925, self-employed - 819, government job - 657. The distribution of various work categories among the persons in the sample was revealed by this visualization.

Count plot for Residence type



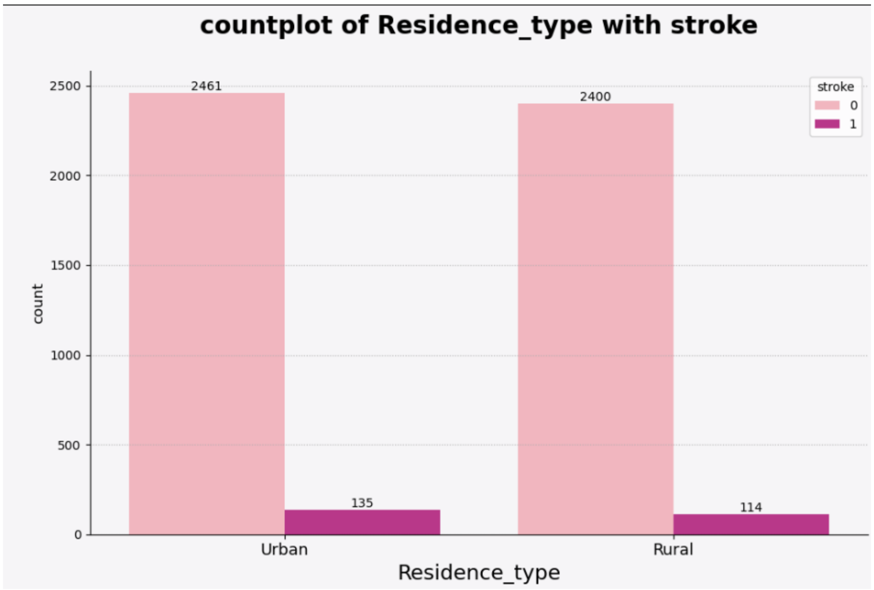
The distribution of the habitation type column was shown, and the results showed that 2514 cases were classed as rural and 2596 cases as urban. We were able to comprehend the distribution of habitation types in the dataset and how they can affect the incidence of strokes thanks to this depiction.

Stroke patients by gender.



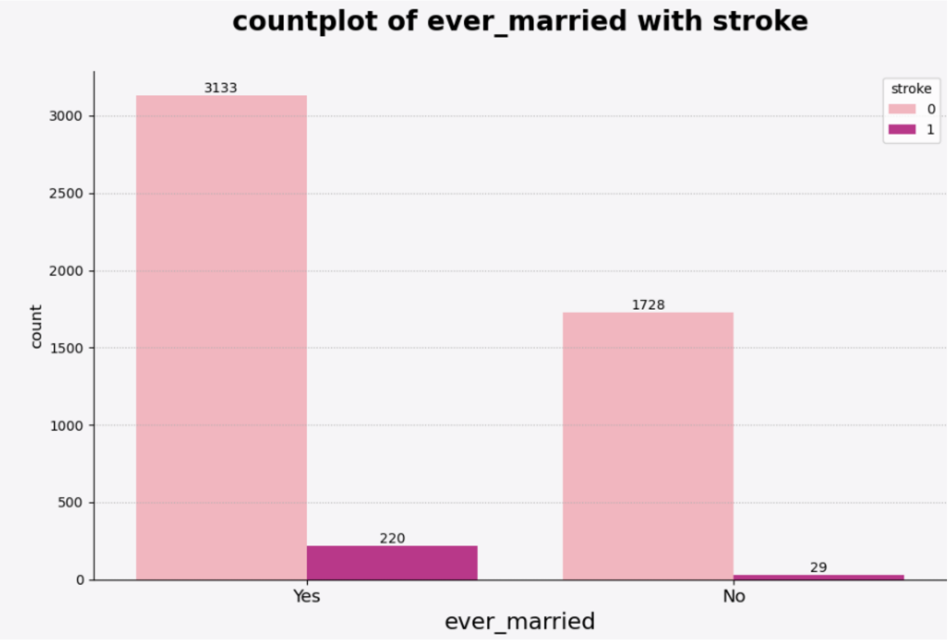
To compare the distribution of stroke events by gender, we made a count plot. We discovered that there were 2007 cases of no strokes and 108 cases of strokes among males. There were 141 cases of strokes and 2853 cases of no strokes among females. We were able to examine the correlation between gender and the number of strokes in the dataset thanks to this graphic.

Stroke patients by Residence type.



To investigate the distribution of stroke events according to habitation type, we made a count plot. We discovered that among those living in cities, there were 135 cases of stroke and 2461 cases of no stroke. There were 114 cases of strokes and 2400 cases of no strokes among the people living in rural areas. We were better able to comprehend the connection between the dataset's habitation type and stroke occurrences thanks to this representation.

Stroke patients by Martial Status

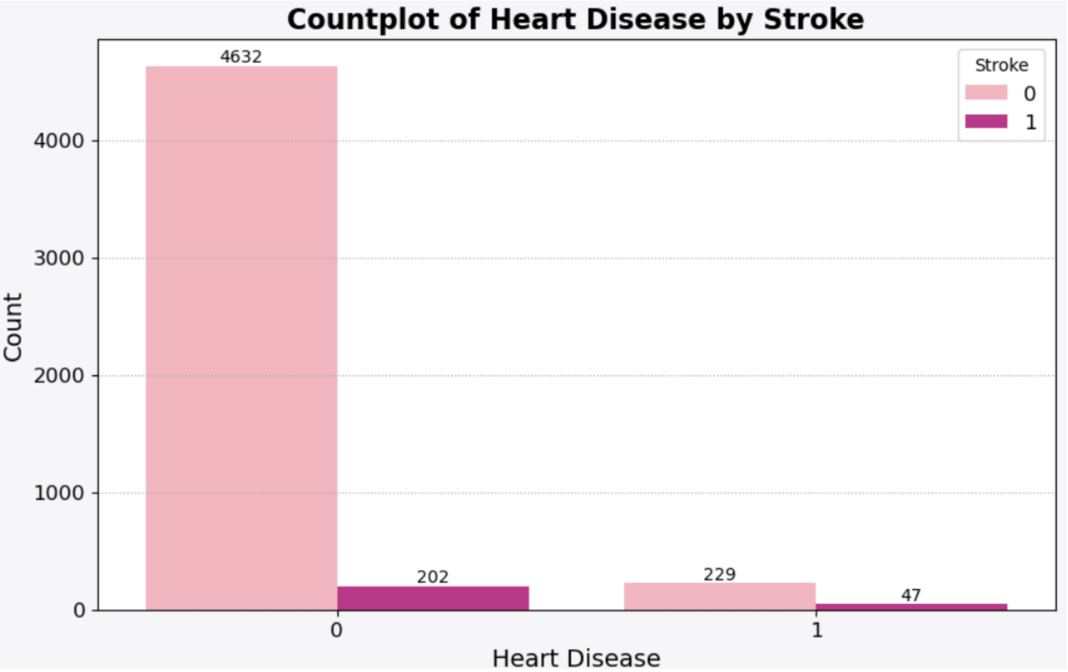


We made a count plot in our earlier study to show how the distribution of stroke cases varied according to marital status. The counts shown below are what we discovered:

- There were 220 cases of strokes and 3133 cases of no strokes among married people.
- There were 29 cases of strokes and 1728 cases of no strokes among the singles.

We were able to investigate the connection between the dataset's marital status and stroke occurrences thanks to this graphic.

Stroke patients by heart disease.

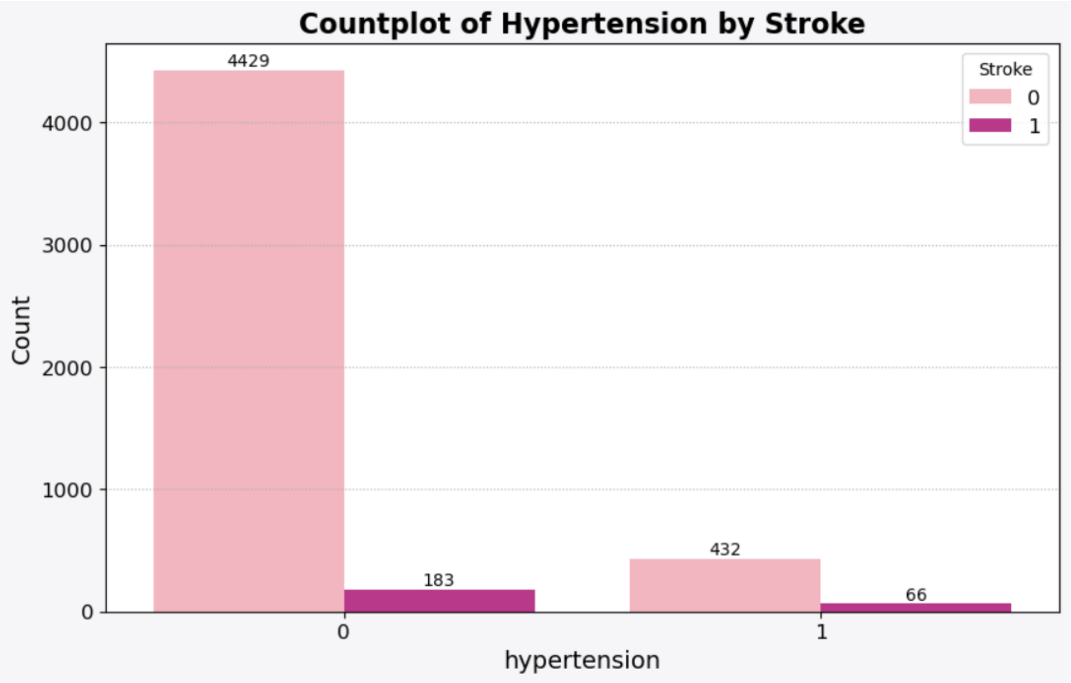


To see the distribution of stroke cases according to the existence or absence of heart disease, we made a count plot. The counts shown below are what we discovered:

- There were 202 cases of strokes and 4632 cases of no strokes among people without heart disease.
- There were 47 cases of strokes and 229 cases of no strokes among patients with heart disease.

We were better able to comprehend the connection between heart disease and stroke incidence in the dataset thanks to this representation.

Stroke patients by Hypertension.

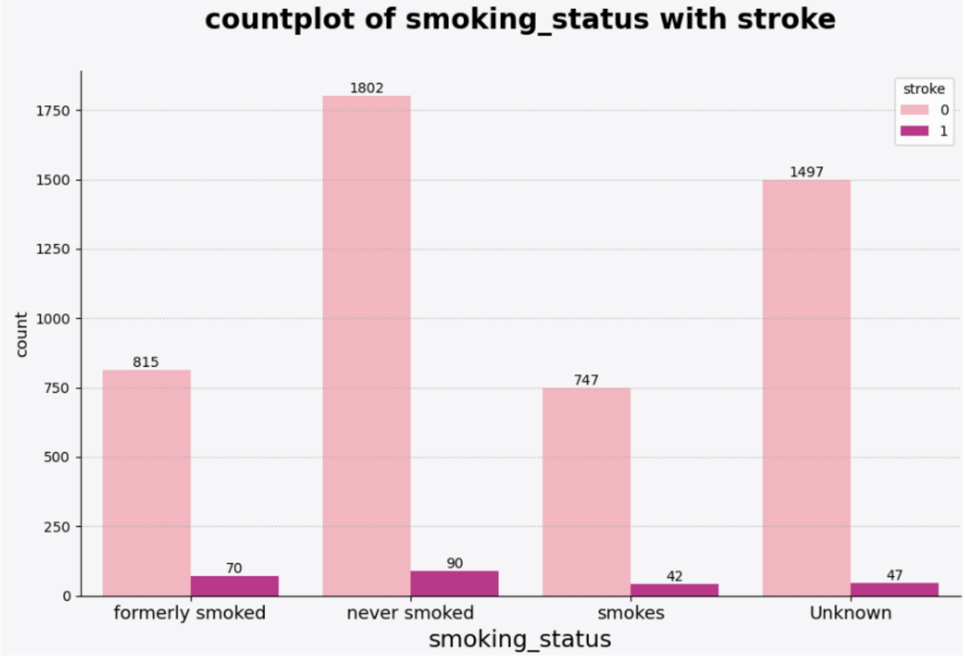


To see the distribution of stroke cases according to the existence or absence of hypertension, we made a count plot. The counts shown below are what we discovered:

- There were 183 cases of stroke and 4429 cases of no stroke among people without hypertension.
- There were 432 cases of strokes and 66 cases of strokes among individuals with hypertension.

We were able to investigate the connection between the dataset's occurrences of stroke and hypertension thanks to this visualization.

Stroke patients by smoking status.

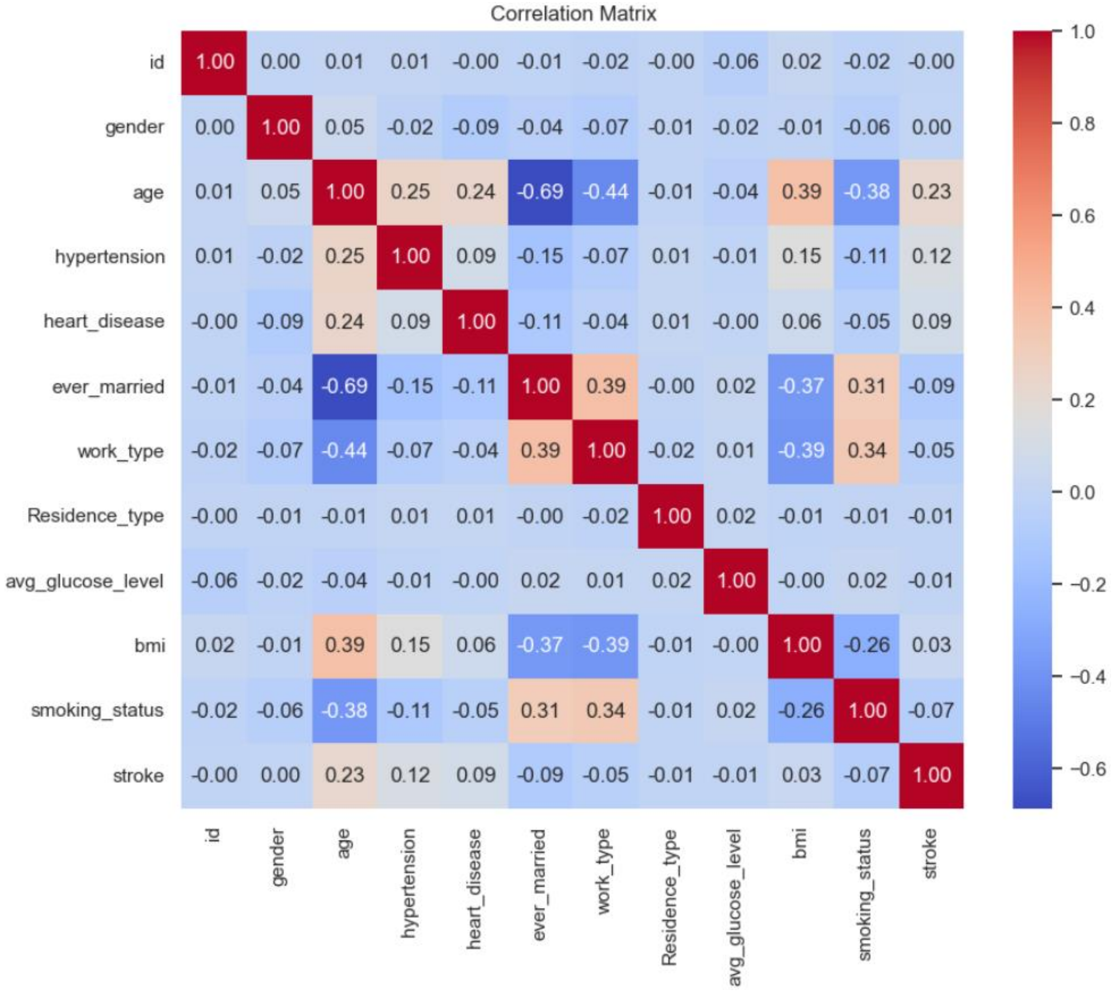


To see how the distribution of stroke cases varied according to smoking status, I made a countplot. The counts shown below are what we discovered:

- There were 815 cases of no strokes and 70 cases of strokes among those who had smoked in the past.
- There were 1802 cases of no strokes and 90 cases of strokes among individuals who never smoked.
- There were 42 cases of stroke and 747 cases of no stroke among smokers.
- There were 47 incidences of strokes and 1947 instances of no strokes among people whose smoking status was unclear.

This graphic shed light on the relationship between smoking status and the number of strokes in the dataset.

Correlation Analysis



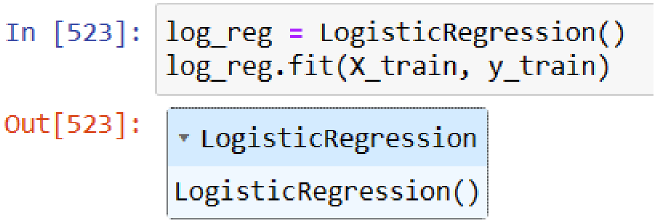
The heatmap's correlation values suggest that there exists a degree of association between all the features and the target variable (stroke), albeit at varying intensities. Age has the highest association (0.23) with stroke, but there are also moderate relationships seen with other variables, including hypertension, heart disease, work type, bmi, and smoking status. On the other hand, characteristics such as Residence_type and avg_glucose_level show less of a relationship with stroke.

Correlation values displayed in the heatmap, we may want to remove Residence_type and avg_glucose_level from our analysis, as their associations with the target variable are not as strong. But it's important to recognize that domain knowledge should also be taken into consideration, and correlation is only one indicator of feature value. As such, you must carefully consider how removing these features may affect your model's predictive capability.

Regression analysis

A statistical technique for examining the relationship between one or more independent variables and a dependent variable is regression analysis. Regression analysis enables us to comprehend the relationship between changes in the independent variables and changes in the dependent variable in the context of predictive modeling. Regression analysis was used in this research to look into the link between a number of predictor variables and the dependent variable—the incidence of strokes.

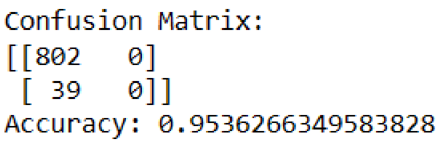
We obtained pertinent predictor factors such age, gender, hypertension, heart disease, marital status, work type, BMI, and smoking status after preprocessing our dataset to eliminate irrelevant columns ('id', 'Residence_type', and 'avg_glucose_level'). The occurrence of strokes, which was the dependent variable in our analysis, was then predicted using these predictor factors.



```
In [523]: log_reg = LogisticRegression()
          log_reg.fit(X_train, y_train)

Out[523]: ▾ LogisticRegression
          LogisticRegression()
```

We used logistic regression, a kind of regression analysis appropriate for problems involving binary classification, to simulate the likelihood of a stroke event according to the predictor factors. Using a logistic function fitted to the observed data, logistic regression estimates the likelihood of an event occurring (in this case, a stroke).

The training and testing sets of the dataset were separated in order to assess the effectiveness of the logistic regression model. The testing set was used to evaluate the model's predicted performance after it had been trained using the training set.



```
Confusion Matrix:
[[802   0]
 [ 39   0]]
Accuracy: 0.9536266349583828
```

Following the logistic regression model's training, we assessed its effectiveness using a number of measures, including the confusion matrix and accuracy score. Based on the presented predictor variables, these metrics provide insights into the model's performance in forecasting the occurrence of strokes.

All things considered, regression analysis helped us pinpoint important stroke occurrence factors and evaluate how each one contributed to the predictive model separately. We can reduce the risk factors linked to strokes by making educated judgments and treatments based on our understanding of the association between predictor variables and stroke occurrence, which will eventually enhance healthcare outcomes. The accuracy of the model is 0.953626, indicating a high level of performance in predicting stroke occurrences.

Conclusion

In conclusion, our thorough investigation and analysis of the Stroke Prediction Dataset using Python have provided valuable insights into the dynamics of stroke occurrences and associated risk factors. Through various stages including data loading and exploration, data cleaning and preprocessing, descriptive analysis, data visualization, correlation analysis, and regression analysis, we have delved into the dataset to uncover patterns, relationships, and predictive models related to strokes.

We began by loading the dataset and exploring its structure, identifying key features such as age, gender, hypertension, heart disease, and more. We addressed challenges such as missing values, outliers, and duplicate entries, ensuring the integrity and reliability of our analysis. Descriptive analysis revealed central tendencies, measures of position, and the spread of variables, providing a comprehensive understanding of the dataset's characteristics.

Through data visualization, we visually represented the distribution of stroke occurrences across different variables such as age, gender, BMI, and smoking status, allowing for a deeper comprehension of their impact on stroke risk. Correlation analysis further elucidated the relationships between variables, highlighting factors with stronger associations to stroke incidence.

Finally, regression analysis facilitated the development of predictive models, allowing us to estimate the likelihood of stroke occurrence based on various predictor variables. Our logistic regression model demonstrated high accuracy in predicting stroke occurrences, enabling informed decision-making and interventions to mitigate stroke risks.

Overall, our analysis contributes to the broader understanding of stroke dynamics and provides valuable insights for healthcare practitioners, policymakers, and researchers in developing targeted interventions and preventive strategies. By leveraging the power of data science and Python analytics, we strive towards reducing the burden of stroke-related disabilities and fatalities, ultimately improving public health outcomes.

Analysis project Codes- [1]

Contribution of the Project

Lanka Pathmakumara	10899186	1) Conducted data cleaning and preprocessing tasks. 2) steps taken to clean the dataset, detailing methods for handling missing values and outliers. 3) Performed descriptive analysis. 4) Conducted correlation analysis to explore relationships between variables. 5) Regression model 6) Documentation.
Ponnahennadige Dias	10899285	1) Led the initial data loading and exploration phase. 2) Documentation
Bathala Wicramasinhe	10899497	1) Documentation
Gabbalage Dilshan	10899287	1) Documentation
Balasuriya Balasuriya	10899180	1) Documentation

References

[1] H. Chandeepa. [Online]. Available: https://drive.google.com/file/d/1RN1Z_M0Ltp2POXPjCvjTrEqxoHIV9be6/view?usp=sharing.

[2] Kaggle, “Stroke Prediction Dataset,” [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data.