Stroke Prediction Machine Learning Project

By: Hellen Amimo Otieno

Hellenamimo72@gmail.com

Table of Contents

TASK 1: DESCRIPTIVE ANALYSIS OF THE STROKE DATASET

1.1 DATA OVERVIEW

1.1.1 Data Shape:

Rows	The stroke dataset consists of 5110 rows representing patient information.
Columns	There are 12 columns. Each patient was assigned an ID, and data was recorded about the patient's gender, age, hypertension status, marital status, work type, residence type, average glucose level, BMI, smoking status, and stroke status.

1.1.2 Data Types:

Attribute	Data Type
Id	Int64
Gender	Object
Age	Float64
Hypertension	Int64
Heart Disease	Int64
Ever Married	Object
Work Type	Object
Residence Type	Object
Average Glucose Level	Float64
BMI	Float64
Smoking status	Object
Stroke	Int64
Based on the data types, the attributes are divided into three categories, integer (4), object (5), and float (3). To enhance the interpretability of the dataset, binary numeric columns (hypertension, heart disease, and stroke) were mapped to categorical labels ("Yes" and "No"). This transformation ensures that the dataset is more accessible to non-technical audiences and that visualizations and group-based analyses are easier to interpret.	

1.1.3 Data types after mapping:

Attribute	Data Type
Id	Int64
Gender	Object
Age	Float64
Hypertension	Object
Heart Disease	Object
Ever Married	Object
Work Type	Object
Residence Type	Object
Average Glucose Level	Float64
BMI	Float64
Smoking status	Object
Stroke	Object

1.2 BASIC STATISTICS FOR NUMERICAL ATTRIBUTES

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

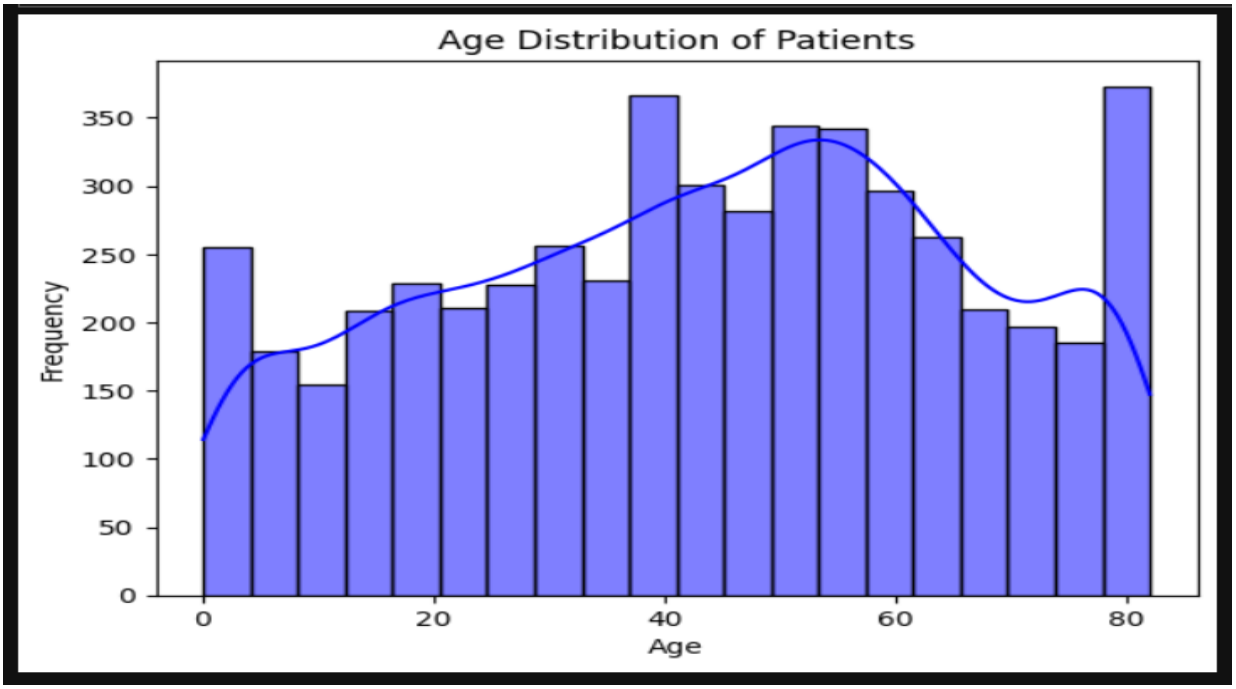
1.2.1 ID

The ID column has no null values. The other basic statistics of ID cannot be interpreted since the ID is a unique identifier for every patient in the dataset. Still, it does not hold any meaningful information about the patient.

1.2.2 AGE

	Interpretation
Count	The age column has 5110 entries, meaning that it has no null values and that the age of every patient was recorded.
Mean	The average age of the patients in the dataset is 43 years, which indicates that the dataset predominantly represents a middle-aged population.
Std	A patient's age in this dataset will likely range between 20 and 65. The dataset focuses on a younger to middle-aged population. It targets working-age adults and does not capture the experience of older adults.
Min	The youngest patient in the dataset is almost one month old. The inclusion of such young patients suggests that the dataset captures cases of pediatric strokes, which are distinct from adult strokes in terms of causes and outcomes.
25%	25% of the patients in the dataset are either 25 years old or below 25 years old. It's important to analyze whether this younger group has distinct stroke risk factors compared to older patients, such as lifestyle choices, family history, or pre-existing health conditions.
50%	50% of the patients in the dataset are either 45 years old or below 45 years. This indicates that half of the patients are in the middle-aged or younger age group.
75%	75% of the patients are either 61 years old or below. This suggests that a significant number of patients in this dataset are in the age group where stroke risk starts to rise.
Max	The oldest patient in this dataset is 82 years old. This reflects that the dataset includes individuals in the elderly age group, who may be more vulnerable to stroke.

Histogram showing the age distribution of patients

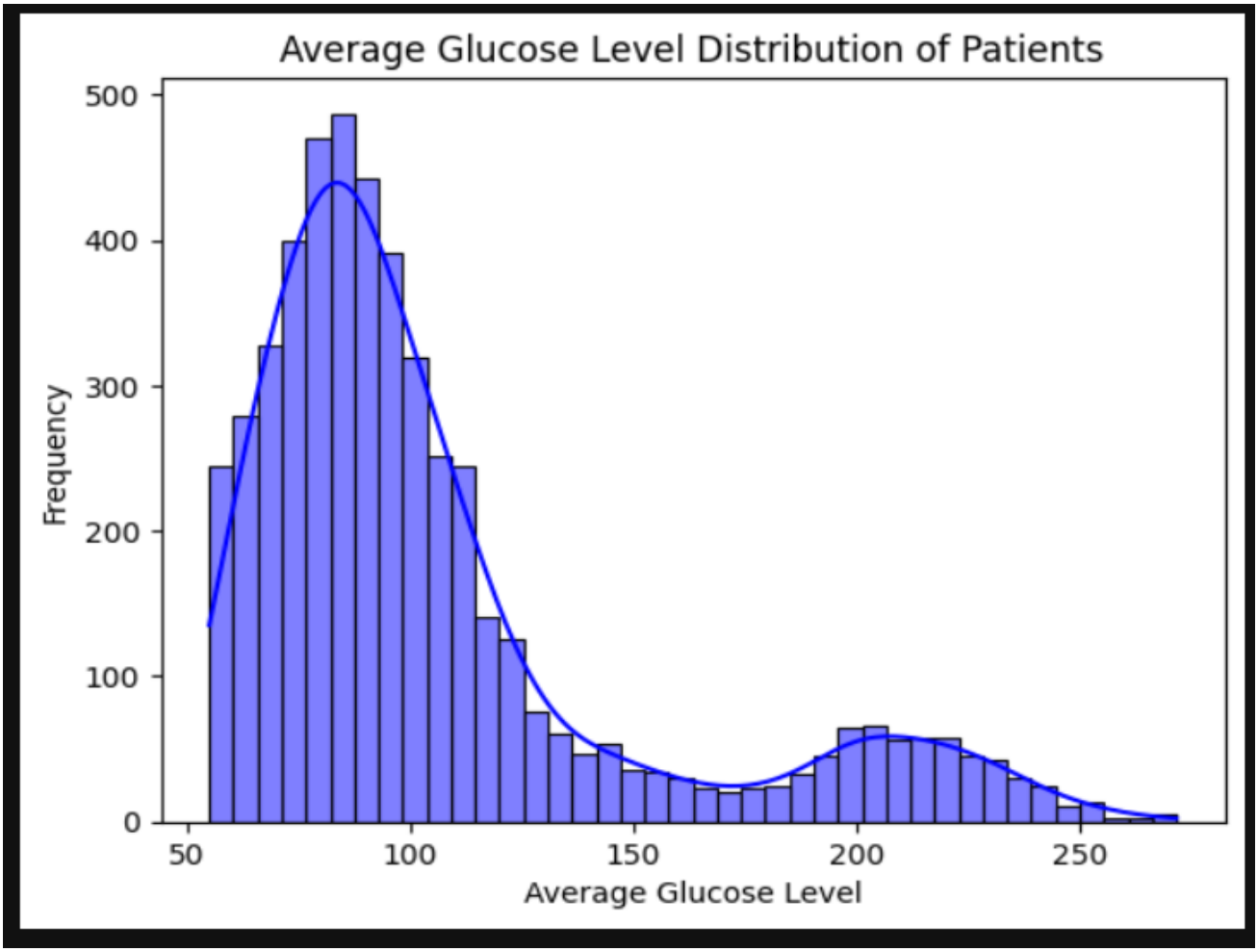


	Interpretation
Skewness	The age histogram is slightly skewed to the left (long tail on the left side), which suggests that there are more older patients in the dataset. Understanding the age distribution helps us see if certain age groups are more prevalent in the dataset, which could be relevant for health analysis.
Range	The width of the histogram indicates a large age range, suggesting that the dataset includes both younger and older patients.
Central Tendency	The tallest bar indicates that the majority of the patients are between 76 and 80 years old, indicating that a significant proportion of patients fall within this range. This peak suggests that elderly individuals are well-represented in the dataset, potentially due to their higher risk of stroke or other health conditions.

1.2.3 AVERAGE GLUCOSE LEVEL

	Interpretation
Count	The average glucose level column has 5110 entries, meaning that it has no null values and that the average glucose level of every patient was recorded.
Mean	The average glucose level of the patients is 106.2 which indicates a population with slightly elevated blood sugar, which aligns with known risk factors for stroke.
Std	The average glucose level of a patient in this dataset is likely to range between 60.9 and 151.4, covering both low and high blood sugar levels. This shows that the patients in the dataset have a mix of blood sugar conditions, with some possibly having low blood sugar and others having high blood sugar.
Min	The lowest average glucose level recorded is 55.1. Low glucose levels in the dataset may indicate that some patients have underlying conditions, such as diabetes management issues.
25%	25% of the patients in the dataset have an average glucose level of 77.24 or less. This suggests that 25% of the patients in the dataset maintain healthy blood sugar levels, which is important for overall health and stroke prevention.
50%	50% of the patients either have an average glucose level of 91.9 or below. This highlights that half of the patients have glucose levels within a healthy range, which could be protective against stroke.
75%	75% of the patients either have an average glucose level of 114.1 or below, which is slightly above normal levels. This suggests that many patients in the dataset could be at risk for health problems related to high blood sugar, which in turn could increase their risk of stroke
Max	The highest average glucose level that was recorded was 271.7. This indicates that some patients in the dataset may be at a significantly higher risk for stroke due to uncontrolled diabetes or other glucose-related health issues.

Histogram showing the distribution of Average Glucose Level of patients



	Interpretation
Skewness	The histogram of average glucose levels shows a right-skewed distribution, where most patients have glucose levels on the lower end, while a smaller proportion exhibits significantly higher levels. This indicates that the dataset contains outliers or extreme values, which could represent patients with diabetes or other health complications.
Range	The range of the average glucose level values suggests that the dataset includes patients with diverse health conditions, from those with normal glucose levels to those with extreme values due to underlying medical issues.
Central Tendency	The tallest bar indicates that the most frequently occurring average glucose level in the dataset is 93.88, indicating that many patients have glucose levels within the normal range. However, the overall distribution is skewed to the right, suggesting the presence of patients with elevated glucose levels.

Box-Plot Showing Outliers in Average Glucose Level

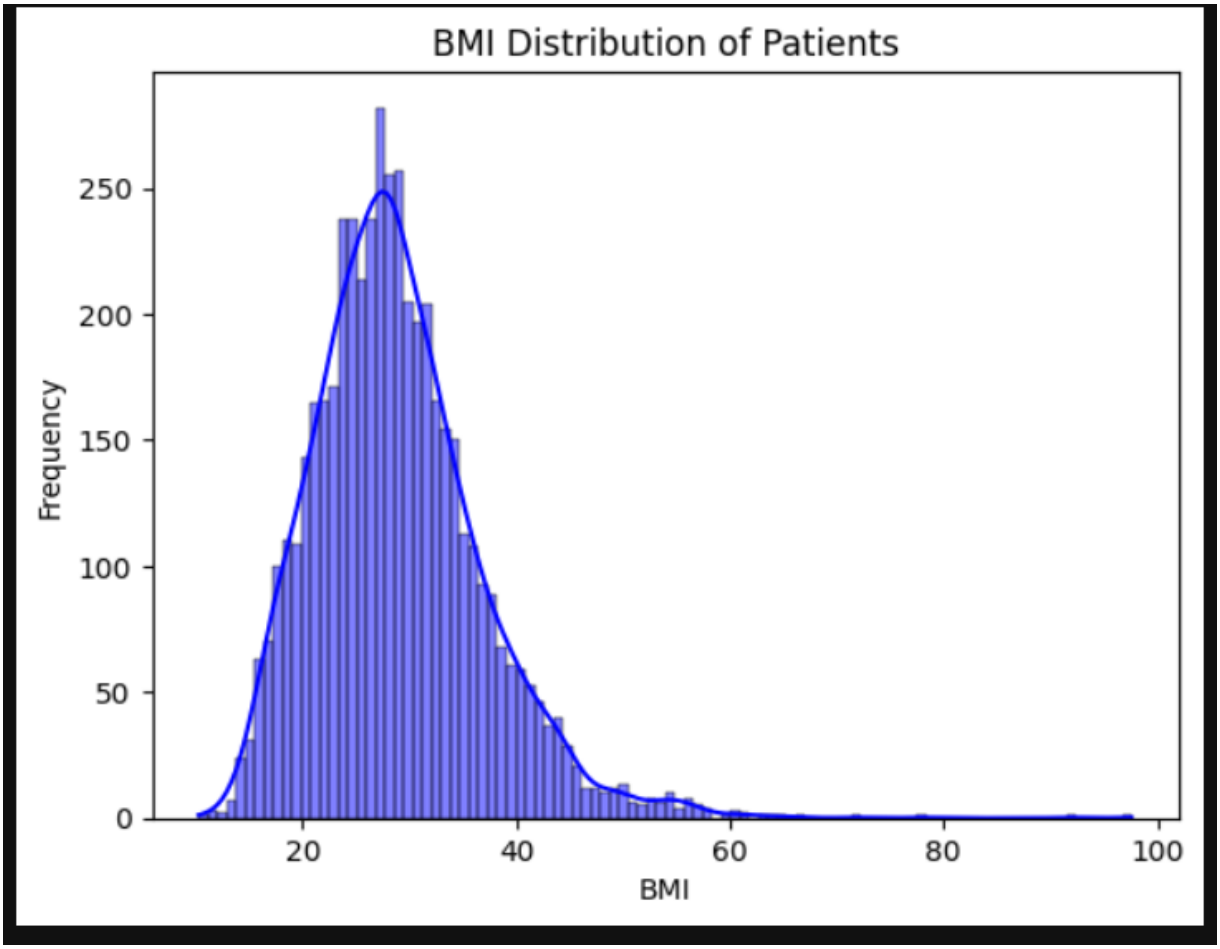
The interquartile range (IQR) is:
IQR= Q3 − Q1= 114.09 - 77.25 = 36.84
Upper Whisker Threshold:
Q3 + 1.5 × IQR = 114.09 + (1.5 × 36.84) = 114.09 + 55.26 = 169.35
Values above 169.35 are considered outliers.
Lower Whisker Threshold:
Q1 − 1.5 × IQR = 77.25 - (1.5 × 36.84) = 77.25 - 55.26 = 21.99

The average glucose level box plot shows a **median of 91.89 mg/dL**, with an **IQR from 77.25 to 114.09 mg/dL**. While most patients have glucose levels within the normal to slightly elevated range, **outliers above 169.35 mg/dL** indicate extremely high blood sugar in some individuals, which may correlate with diabetes and an increased stroke risk. The broad range of glucose levels emphasizes the need to address high blood sugar levels (hyperglycemia) and low blood sugar(hypoglycemia) when analyzing stroke risk factors in this population.

1.2.4 BMI

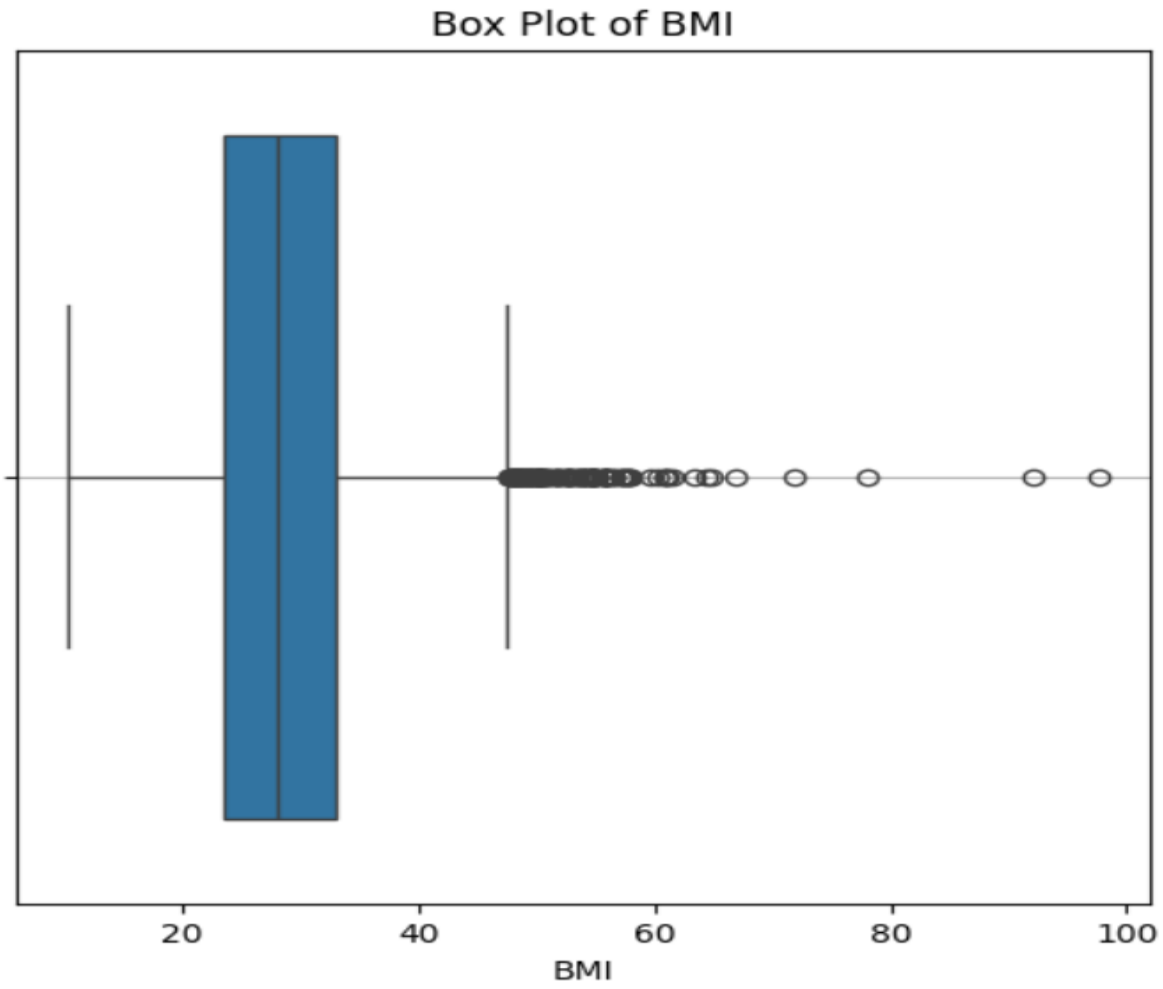
	Interpretation
Count	The BMI column has 4909 entries, meaning that only 4909 patients' BMIs were recorded, while the BMIs of 201 patients were not recorded.
Mean	The average BMI of the patients is 29, suggesting that on average, the patients fall in the overweight category. A BMI of 18.5 to 24.9 is generally considered healthy, 25 to 29.9 is categorized as overweight, and 30 or higher is classified as obese.
Std	The BMI of a patient in this dataset is likely to range between 21 and 37. In this dataset, the lower end of the range (21) suggests patients with healthy weights, while the upper end (37) indicates obesity.
Min	The lowest BMI recorded is 10.3. This suggests severe undernutrition or a serious health condition. This level is much lower than typical BMI values and could indicate that the patient is suffering from a condition like anorexia, severe malnutrition, or other serious medical issues.
25%	25% of the patients have a BMI of 23.5 or less which falls within the healthy weight range. This suggests that a quarter of the patients are maintaining a healthy body weight, which is generally associated with a lower risk of stroke.
50%	50% of the patients either have a BMI of 28.1 or below. This value falls in the overweight category. While this is not considered obese, it still suggests that many patients may have excess body weight, which is associated with a higher risk of stroke.
75%	75% of the patients either have a BMI of 33.1 or below, which is considered obese. Obesity is a known risk factor for stroke and other health conditions like high blood pressure and diabetes. This suggests that many patients in the dataset may be at a higher risk for stroke due to their weight.
Max	The highest BMI recorded is 97.6 which indicates that there may be patients in this dataset who are at a very high risk of stroke due to extreme obesity

Histogram showing BMI Distribution of Patients



	Interpretation
Skewness	The histogram of average glucose levels shows a right-skewed distribution, where most patients have glucose levels on the lower end, while a smaller proportion exhibits significantly higher levels. This indicates that the dataset contains outliers or extreme values, which could represent patients with diabetes or other health complications.
Range	The range of the average glucose level values suggests that the dataset includes patients with diverse health conditions, from those with normal glucose levels to those with extreme values due to underlying medical issues.
Central Tendency	The tallest bar indicates that the most frequently occurring average glucose level in the dataset is 93.88, indicating that many patients have glucose levels within the normal range. However, the overall distribution is skewed to the right, suggesting the presence of patients with elevated glucose levels

Box-Plot Showing Outliers in BMI



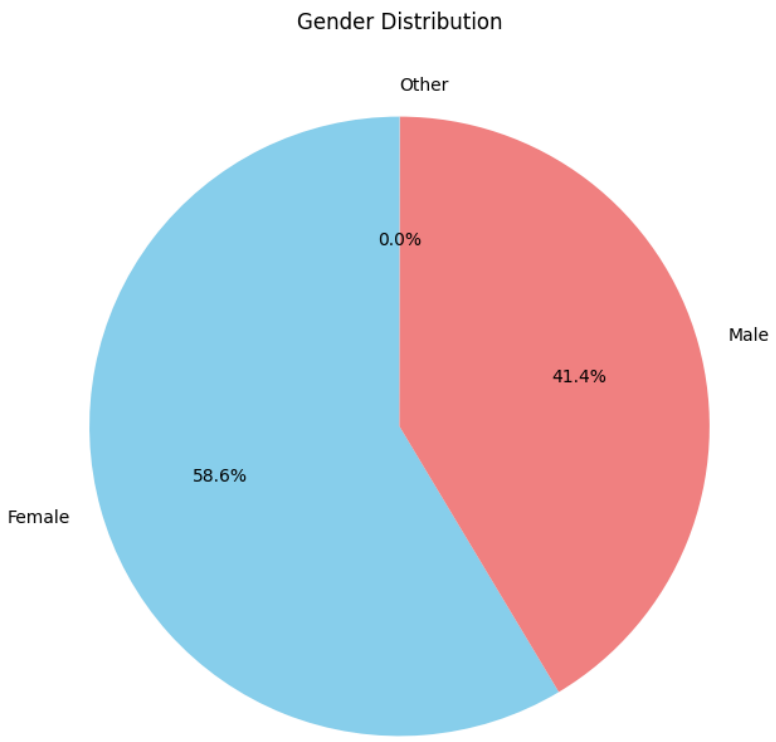
The interquartile range (IQR) is: IQR= Q3 − Q1= 33.1 − 23.5 = 9.6 **Upper Whisker Threshold:** Q3 + 1.5 × IQR = 33.1 + (1.5×9.6) = 33.1 + 14.4 = 47.5 Values above 47.5 are considered outliers. **Lower Whisker Threshold:** Q1 − 1.5 × IQR = 23.5 − (1.5×9.6) = 23.5 − 14.4 = 9.1
The BMI box plot illustrates the distribution of BMIs among stroke patients. With an IQR spanning from 23.5 to 33.1 and a median of 28.1, it indicates that most patients are overweight. Outliers above **47.5** highlight extreme obesity cases, which could be critical in understanding stroke risk factors. These outliers could also signal the need for a closer look at data quality or specific patient characteristics.

1.3 BASIC STATISTICS FOR CATEGORICAL ATTRIBUTES

	gender	hypertension	heart_disease	ever_married	work_type	Residence_type	smoking_status	stroke
count	5110	5110	5110	5110	5110	5110	5110	5110
unique	3	2	2	2	5	2	4	2
top	Female	No hypertension	No heart disease	Yes	Private	Urban	never smoked	No stroke
freq	2994	4612	4834	3353	2925	2596	1892	4861

1.3.1 GENDER

Pie chart showing gender distribution of patients

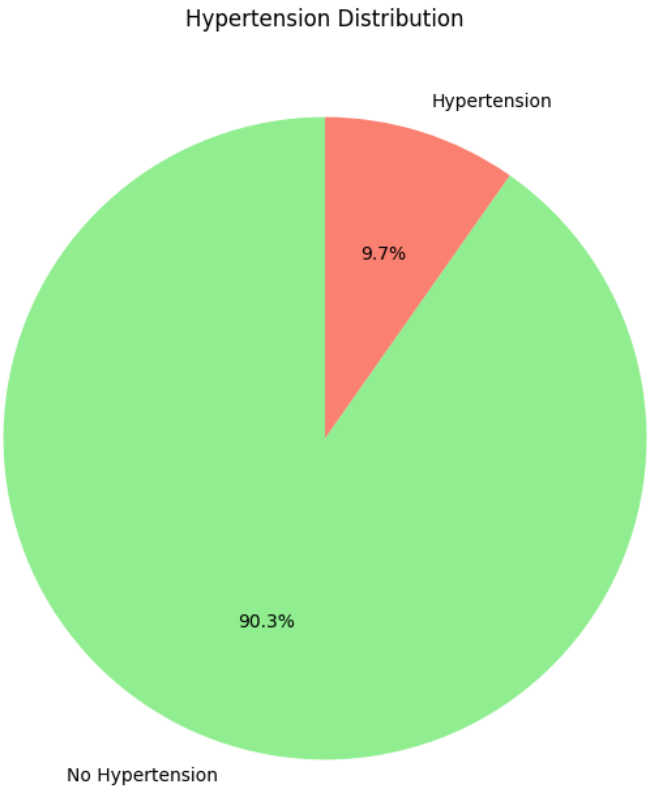


```
gender
Female    2994
Male      2115
Other        1
Name: count, dtype: int64
```

Count = 5110	The gender column has no null values. This means that the gender of every patient was recorded.
Unique	This column has three unique gender categories: Male, Female, or Other. This indicates that the dataset accounts for a broad range of gender identities, including traditional binary categories (Male and Female) and a third category, **Other**, which may represent other gender identities.
Top	There are more female than male patients and "Other" patients in this dataset.
Frequency	The dataset has a larger proportion of females (2,994 instances), followed by males (2,115 instances), and only 1 instance of "Other." This suggests that the dataset is mostly binary in gender representation, with Other being a rare category.

1.3.2 HYPERTENSION

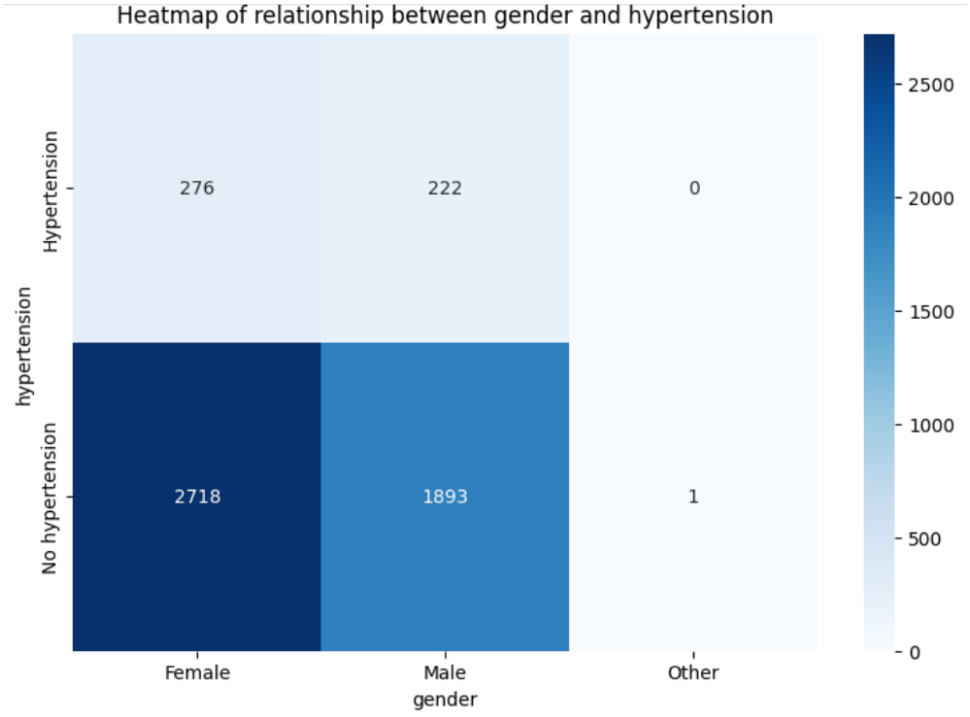
Pie chart showing hypertension distribution



```
hypertension
No hypertension    4612
Hypertension        498
Name: count, dtype: int64
```

Count = 5110	The hypertension column has no null values. This means that the hypertension level of every patient was recorded.
Unique	The hypertension column contains two unique categories: the patient either has hypertension or the patient does not have hypertension.
Top	Patients with no hypertension are the most frequent in this dataset.
Frequency	A large majority of patients in the dataset do not have hypertension, with 4,612 instances (about 90.4% of the dataset). In contrast, only a smaller proportion of patients, 498 instances (around 9.6%), are reported to have hypertension.

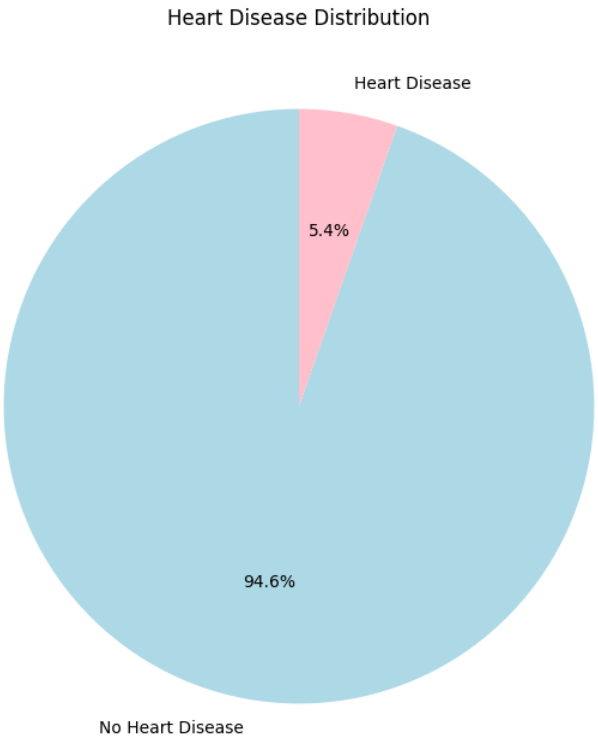
Heatmap showing the relationship between gender and hypertension



The majority of individuals in the dataset, regardless of gender, do not have hypertension. Among those with hypertension, females slightly outnumber males due to their larger population in the dataset. Males appear to have a higher occurrence of hypertension (10.5%) compared to females (9.2%). The 'Other' gender category has only one individual recorded, which limits its contribution to meaningful analysis.

1.3.3 HEART DISEASE

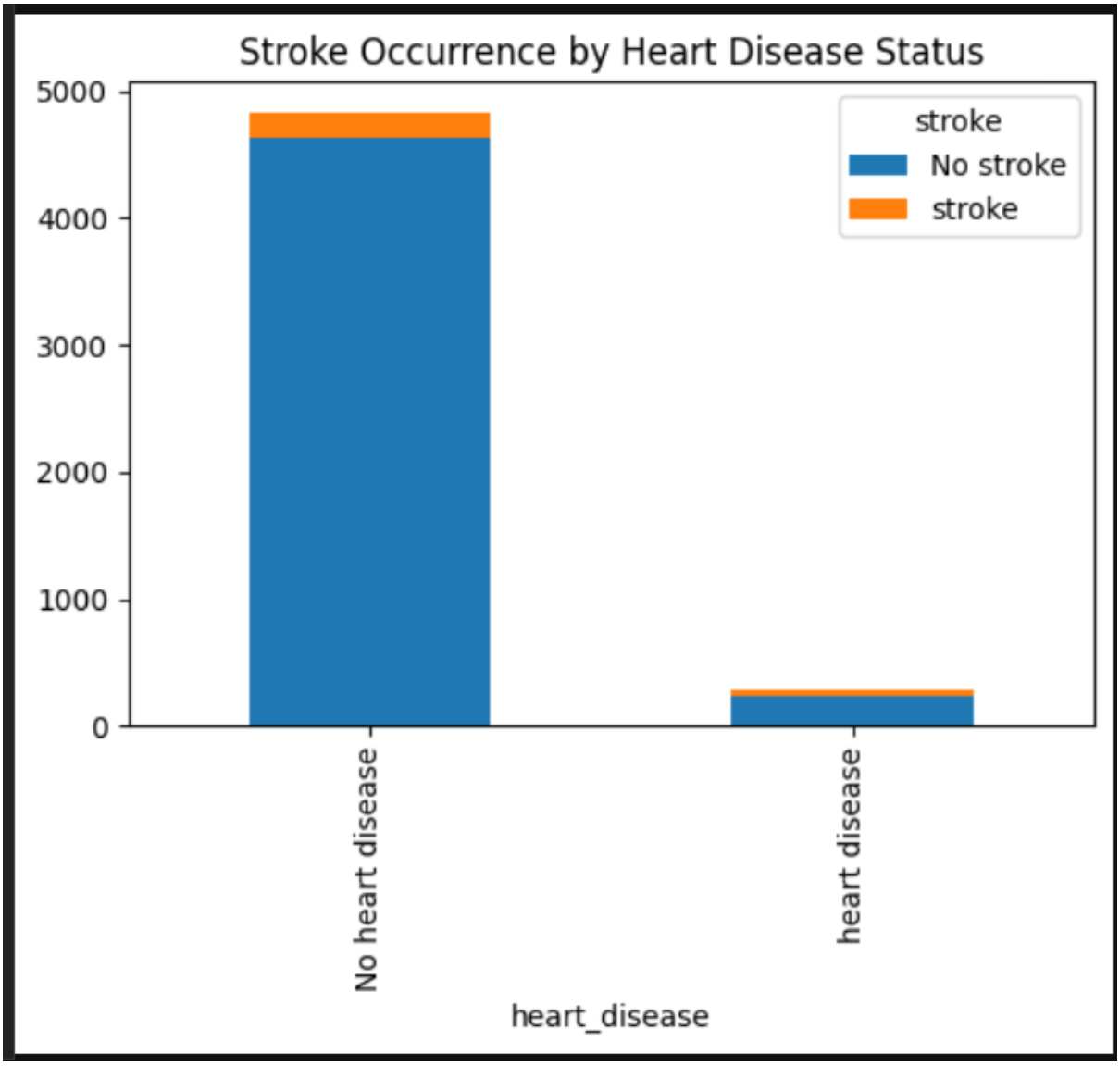
Pie chart showing the distribution of heart disease among the patients




```
heart_disease
No heart disease    4834
heart disease        276
Name: count, dtype: int64
```

Count = 5110	The heart disease column has no null values. This means that the heart disease status of every patient was recorded.
Unique	The heart disease column contains two categories: the patient either has heart disease or the patient does not have heart disease.
Top	Patients with no heart disease are the most frequent in this dataset.
Frequency	The majority of patients in the dataset do not have heart disease, with 4,834 instances (about 94.6% of the dataset). A smaller proportion of patients, 276 instances (around 5.4%), are reported to have heart disease.

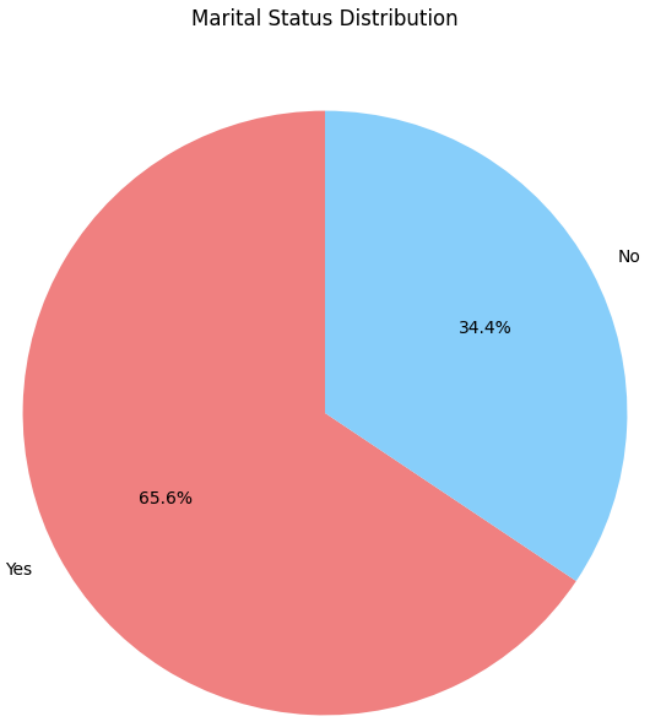
Stacked column chart showing the relationship between heart disease and strokes



The highest proportion of people with strokes is observed in the **no heart disease** category. This suggests that individuals without heart disease may have other underlying conditions or risk factors (such as hypertension, diabetes, or lifestyle factors) that contribute more significantly to the occurrence of strokes. This finding may indicate that heart disease is not the strongest factor driving stroke occurrences in this dataset.

1.3.4 EVER MARRIED

Pie chart showing the distribution of marital status

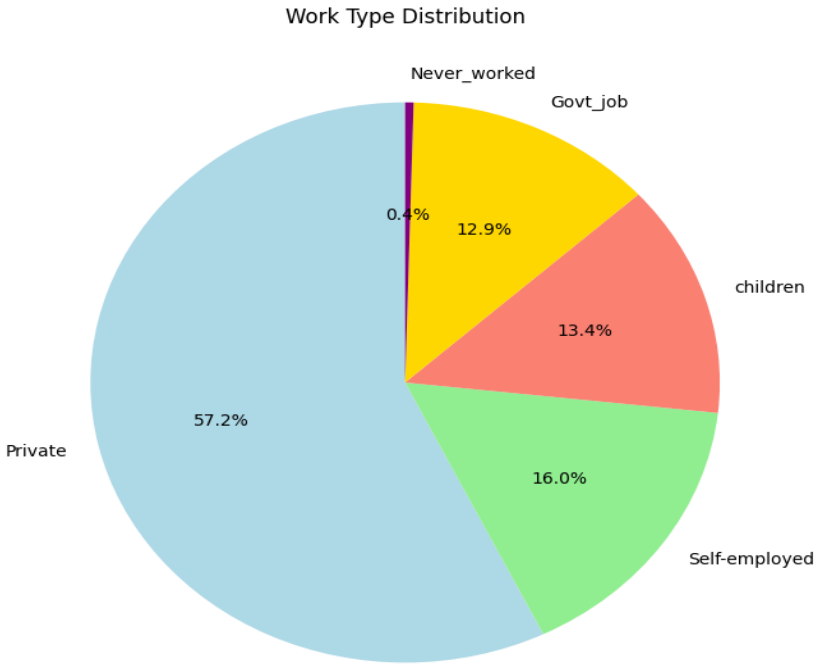


```
ever_married
Yes     3353
No      1757
Name: count, dtype: int64
```

Count = 5110	The ever-married column has no null values. This means that the marital status of every patient was recorded.
Unique	The ever-married column contains two categories: Yes (the patient has ever been married) and No (the patient has never been married).
Top	Patients who have ever been married are the most frequent in this dataset.
Frequency	A majority of patients in the dataset have been married, with 3,353 instances (about 65.6% of the dataset). A smaller proportion of patients, 1,757 instances (around 34.4%), have never been married.

1.3.5 WORK TYPE

Pie chart showing work type distribution

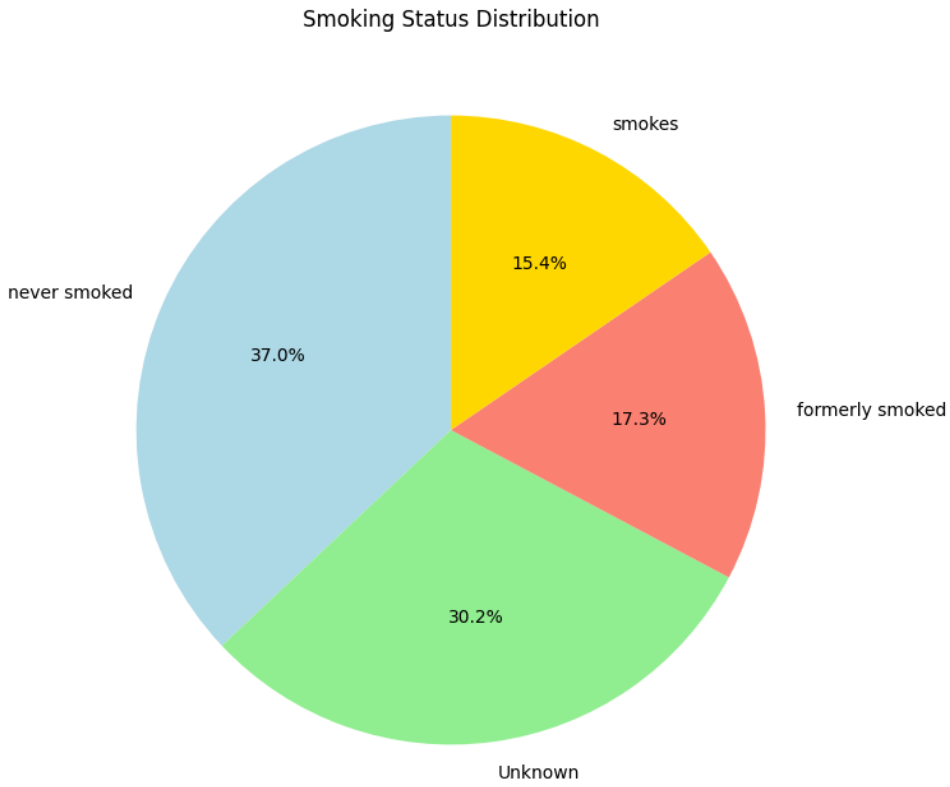



```
work_type
Private          2925
Self-employed     819
children          687
Govt_job          657
Never_worked       22
Name: count, dtype: int64
```

Count = 5110	The work type column has no null values. This means that the work type of every patient was recorded
Unique	The work type column contains five categories: Private, Self-employed, children, Government job, and Never worked.
Top	Patients who work in the private sector are the most frequent in the dataset.
Frequency	The most common work type in the dataset is Private, with 2,925 instances (about 57.2% of the dataset)."Self-employed" patients make up 819 instances (around 16.0%)."Children" is recorded for 687 patients (around 13.4%)."Government job" is recorded for 657 patients (around 12.9%). A very small number of patients, 22 instances (about 0.4%), are listed as Never worked.

1.3.6 SMOKING STATUS

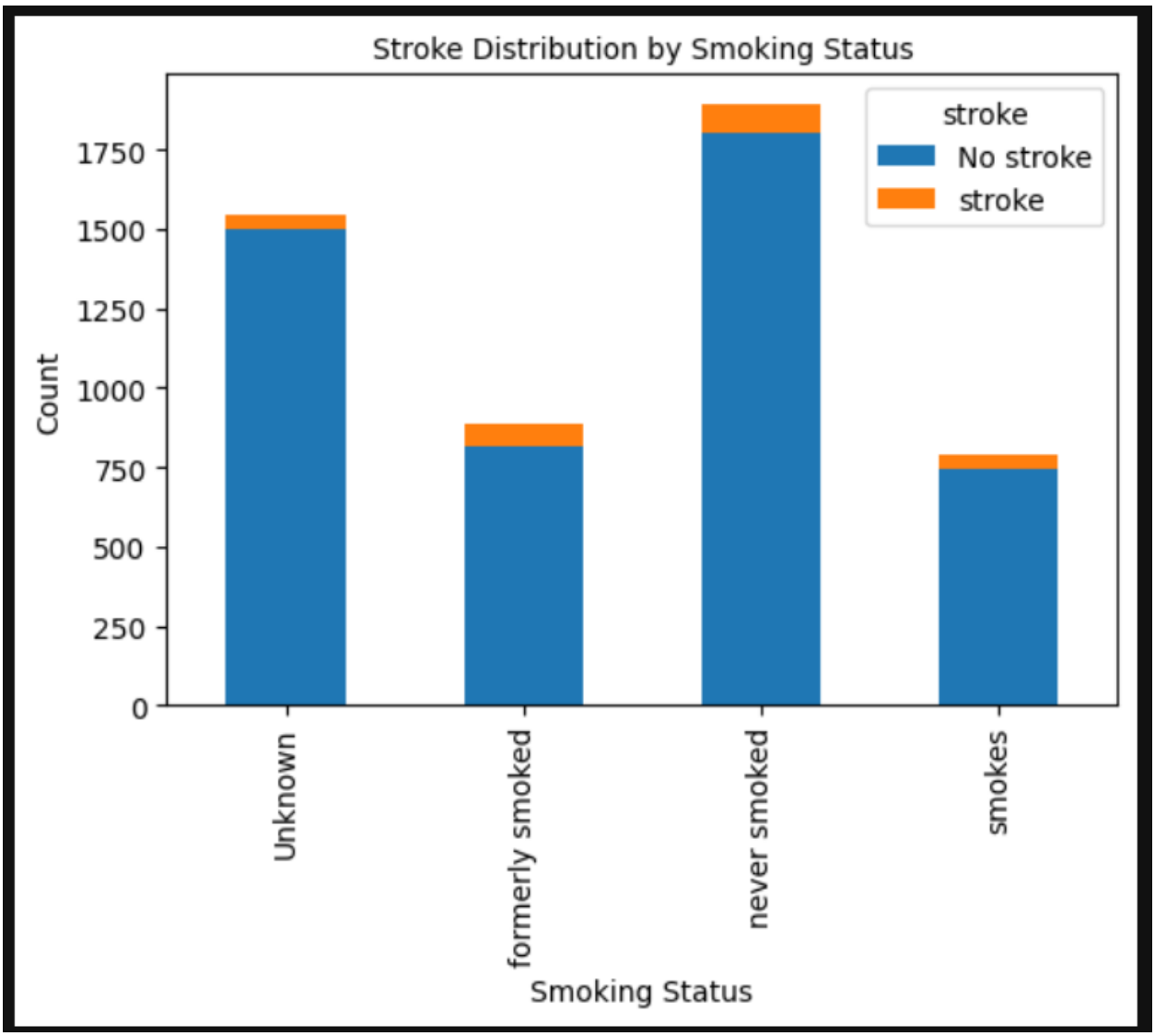
Pie chart showing the smoking status distribution of patients




```
smoking_status
never smoked       1892
Unknown            1544
formerly smoked     885
smokes              789
Name: count, dtype: int64
```

Count = 5110	The smoking status column has no null values. This means that the smoking status of every patient was recorded
Unique	The smoking status column contains four categories: “never smoked”, “unknown”, “formerly smoked”, and “smokes”.
Top	Patients who have never smoked are the most frequent in the dataset.
Frequency	The most common smoking status in the dataset is patients who have never smoked, with 1,892 instances (about 37.0% of the dataset). "Unknown" smoking status is recorded for 1,544 patients (around 30.2%). Patients who used to formerly smoke make up 885 instances (around 17.3%). Patients who smoke are recorded in 789 instances (about 15.4%).

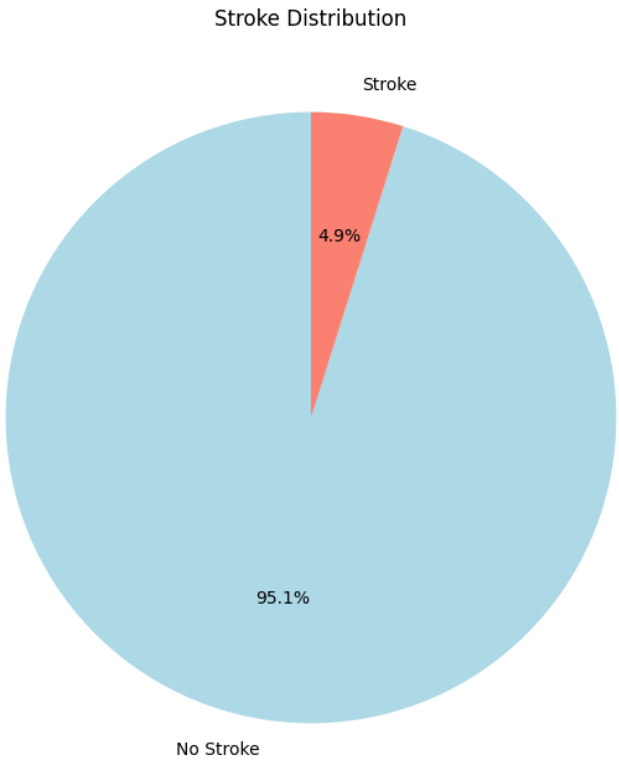
Stacked column chart showing the relationship between Smoking status and Strokes



The proportion of people with strokes is relatively consistent across smoking statuses, with the highest proportion observed in the 'never smoked' category. However, the differences are minimal, suggesting that smoking status alone may not be a strong differentiator in stroke occurrence in this dataset. This observation may indicate the presence of other variables that influence stroke risk more significantly than smoking status.

1.3.7 STROKE

Pie chart showing stroke distribution



```
stroke
No stroke    4861
stroke        249
Name: count, dtype: int64
```

Count = 5110	The stroke status column has no null values. This means that the stroke status of every patient was recorded
Unique	The stroke column contains two categories: “stroke” (patients who have had a stroke) and “no stroke” (patients who have not had a stroke).
Top	Patients who have never had a stroke are the most frequent in this dataset.
Frequency	The majority of patients in the dataset have not had a stroke, with 4,861 instances (about 95.1% of the dataset). A smaller proportion of patients, 249 instances (around 4.9%), have had a stroke.

TASK 2: DATA PREPARATION FOR CLASSIFICATION, REGRESSION AND CLUSTERING

I used the following methods to prepare my data for classification, regression, and clustering models:

2.1: DROPPING NOISY ATTRIBUTES

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

I dropped the ID attribute since it consists of unique identifiers that do not provide any predictive value for the regression, classification, or clustering tasks. Including the ID in the model could introduce unnecessary complexity and risk overfitting, especially in regression and classification models where the ID might accidentally correlate with the target variable. The ID does not help group similar observations for clustering algorithms, as it is an arbitrary label that doesn't reflect any underlying patterns in the data.

2.2: CHECKING FOR OUTLIERS

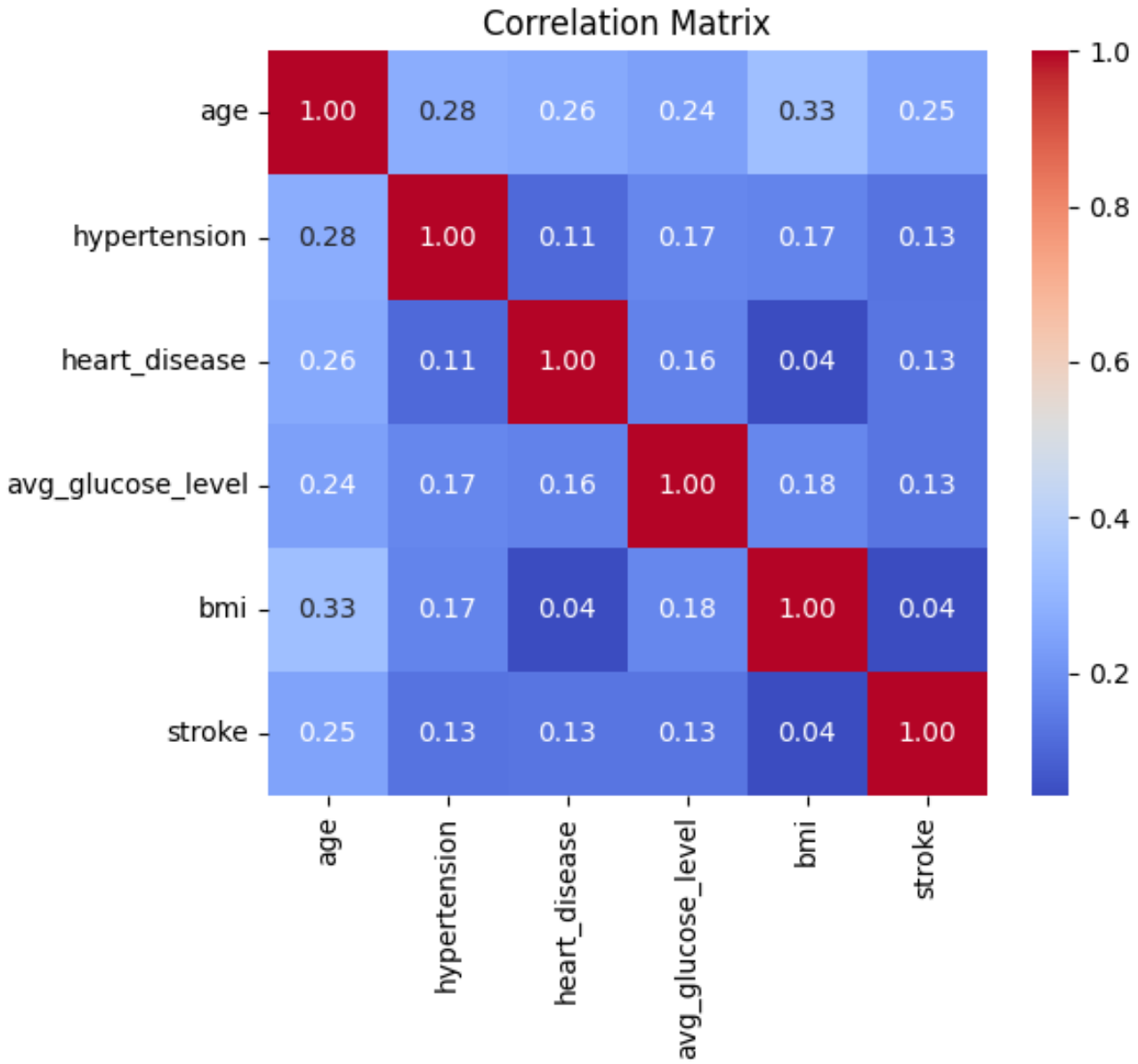
As an extra step, I used the interquartile range (IQR) method (Zafeirelli, S., & Kavroudakis, D. (2024)) which is a measure of statistical dispersion that captures the range within which the central 50% of the data lies.
It is calculated as the difference between the third quartile (Q3) and the first quartile (Q1):
IQR = Q3 − Q1
To detect potential outliers, thresholds are determined using the IQR.
The **upper whisker threshold** is defined as:
Q3 + 1.5 × IQR
The **lower whisker threshold** is defined as:
Q1 − 1.5 × IQR
Data points falling outside these thresholds are considered potential outliers.

Attribute	Outliers	Action taken
Age	No outliers	No action required
Average glucose level	**627** entries with outliers, meaning that their values are higher than 169.35 which is the upper threshold.	I decided to retain these outliers because extreme glucose levels are clinically significant. High or low glucose levels could indicate undiagnosed diabetes or hypoglycemia, both of which are important risk factors for stroke.
BMI	**110** entries with outliers, meaning that their values are higher than 47.5 which is the upper threshold.	I decided to retain outliers in BMI values since they are clinically significant and linked to conditions that affect stroke risk, such as obesity or malnutrition.

I used the frequency count to check for outliers in the categorical features.

Attribute & Frequency count	Outliers	Action taken
Gender Female = 2994 Male = 2115 Other = 1	Other = 1	I decided to drop this observation because the "Other" category contains only a single observation, which is too small to meaningfully contribute to the analysis
Ever Married Yes = 3353 No = 1757	No outliers	No action required
Work Type Private = 2925 Self-employed = 819 Children = 687 Never worked = 22	The "Never worked" category has only 22 observations, which is an outlier relative to the other categories	The category may include students, retirees, or people with disabilities, which could be relevant when analyzing stroke risk or other health conditions. Removing this group could lead to the exclusion of important patterns related to this subgroup.
Residence Type Urban = 2596 Rural = 2514	No outliers	No action required
Hypertension No hypertension = 4612 Hypertension = 498	The no hypertension category has a significantly higher count than hypertension.	This might not be an outlier, but a significant imbalance. I did not drop any outliers in this category.
Heart Disease No heart disease = 4834 Heart disease = 276	The number of people without heart disease is much larger than those with heart disease.	Similar to hypertension, there is a significant imbalance between the two categories that might require attention in modeling.
Smoking Status Never smoked = 1892 Unknown = 1554 Formerly smoked = 885 Smokes = 789	The "Unknown" category with 1544 observations stands out as having more counts than the "Smokes" and "Formerly Smoked" categories.	The "Unknown" category might not be an outlier, but its higher count could suggest unreported data.
Stroke No stroke = 4861 Stroke = 249	There is a significant imbalance between stroke and no stroke. categories. The majority of observations are for individuals without a stroke.	This imbalance might not be an outlier, but rather a class imbalance, which could affect model performance.

2.3: CORRELATION MATRIX



I plotted the correlation matrix above to understand how the attributes in the stroke dataset are related to each other. The lack of strong correlations in the stroke dataset suggests that the features are not linearly related to each other, which can be advantageous for certain models like decision trees or random forests.

2.4: HANDLING MISSING VALUES

Attribute	Missing Values
Gender	0
Age	0
Hypertension	0
Heart Disease	0
Ever married	0
Work type	0
Residence type	0
Average Glucose Level	0
BMI	**201**
Smoking status	0
Stroke	0

The only attribute that had missing values in the dataset was BMI which had 201 missing values.
For **regression, I dropped the rows with BMI missing values**. I did this because regression models require complete data, and missing BMI values constitute only a small portion of the dataset, these rows were dropped to maintain the integrity of the analysis. Additionally, imputing BMI values could introduce bias or distort the variance of this feature.
For **classification and clustering**, **I imputed the missing values with the median** because the median is a robust measure of central tendency that is not affected by outliers, making it an ideal choice for imputation.

2.5: LABEL ENCODING

Before performing classification, regression, and clustering, I transformed all the attributes to numerical data type and this was the result:

Attribute	Data types before label encoding	Data types after label encoding
Gender	Object	Int32
Age	Float64	Float64
Hypertension	Int64	Int64
Heart Disease	Int64	Int64
Ever married	Object	Int32
Work type	Object	Int32
Residence type	Object	Int32
Average glucose level	Float64	Float64
BMI	Float64	Float64
Smoking status	object	Int32
Stroke	Int64	Int64

This step was necessary because most machine learning algorithms require numeric data for processing and computation. Label encoding preserves categorical information, enables mathematical operations, and ensures compatibility with models while avoiding the loss of important features. For ordinal features, it also captures their natural order, making it an efficient and interpretable preprocessing step.

2.6: Splitting the train and test sets and Balancing the "stroke" classes

```
stroke
0    4861
1     249
Name: count, dtype: int64
```

The stroke classes are imbalanced and this would affect the classification process.
To prepare the dataset for classification I split the data into training (75%) and testing (25%) subsets using the train_test_split method.
As an extra step, I applied the **Synthetic Minority Oversampling Technique (SMOTE)** (Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002) which is a popular method used to address class imbalance in datasets before performing classification. It works by generating synthetic samples for the minority class rather than simply duplicating existing samples. This helps create a more balanced dataset and improves the performance of classification models.

The figure below shows the balanced stroke classes.

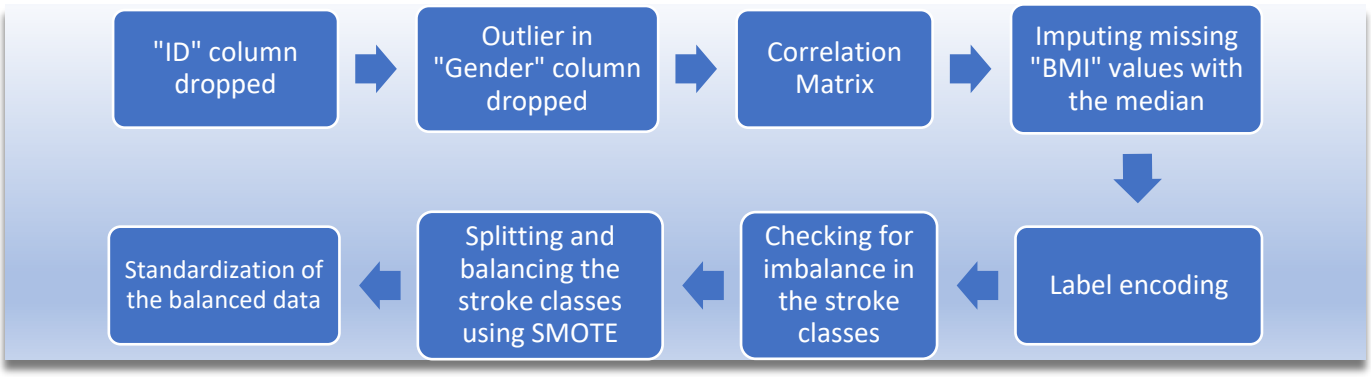
```
Counter({np.int64(0): 3662, np.int64(1): 3662})
```

2.7: STANDARDIZATION USINF Z-SCORE SCALING

Classification	Standardization using Z-Score Scaling was applied to the dataset before classification to ensure all features have a mean of 0 and a standard deviation of 1. The scaler was fitted on the training data to compute the mean and standard deviation, and these parameters were then used to transform both the training and test datasets. This step was crucial for algorithms that rely on distance measures, such as clustering and k-nearest Neighbors, to prevent features with larger scales from dominating the calculations.
Clustering	Before performing clustering, the features were standardized using z-score scaling to ensure they have comparable scales. Each feature was transformed to have a mean of 0 and a standard deviation of 1. Standardization is crucial for clustering algorithms like K-Means, which are sensitive to the scale of features, as it prevents features with larger ranges from dominating the distance calculations.

TASK 3: CLASSIFICATION OF THE 'STROKE' ATTRIBUTE

This was the data preparation pipeline I followed before performing classification.

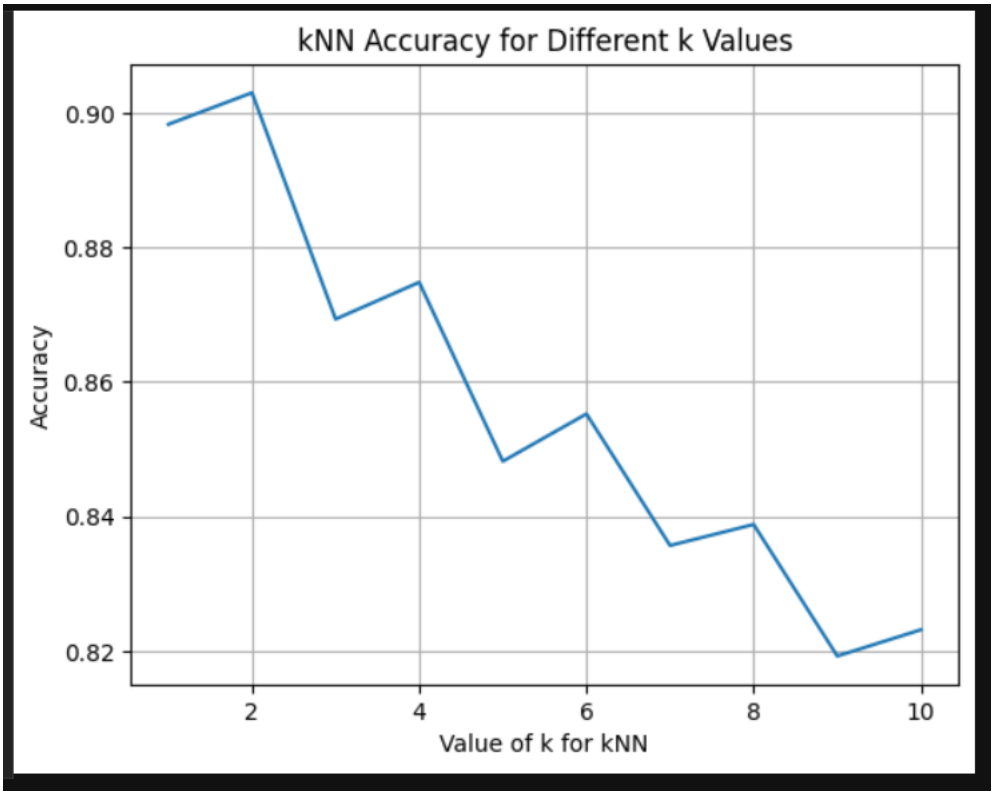


I used the following classification algorithms: kNN, Naïve Bayes, Decision Tree, and XGBoost. The results are as follows:

3.1 k-NEAREST NEIGHBOUR CLASSIFIER

To determine the optimal value for **k** (the number of nearest neighbors considered) in the k-Nearest Neighbors (kNN) algorithm, I tested different values of k and measured the accuracy of the model for each value. I plotted a graph showing the accuracy level of every value of k. From the graph and accuracy values, the optimal value of k is 2.

Value of k	Accuracy
k = 1	0.8982785602503912
k = 2	**0.9029733959311425**
k = 3	0.8693270735524257
k = 4	0.8748043818466353
k = 5	0.8482003129890454
k = 6	0.8552425665101722
k = 7	0.8356807511737089
k = 8	0.838810641627543
k = 9	0.8192488262910798
k = 10	0.8231611893583725



3.1.1 Model 1: k-NN with k = 2

I performed cross-validation with ten splits and k=2 and the accuracy is as follows:

```
[0.95907231 0.97544338 0.97271487 0.97953615 0.97404372 0.97540984
 0.96994536 0.96448087 0.96584699 0.96994536]
Accuracy: 0.97 (+/- 0.01)
```

Below is a summary of the performance metrics:

```
Confusion matrix
[[3486 176]
 [ 39 3623]]

                  precision    recall  f1-score   support

       No stroke       0.99      0.95      0.97      3662
          Stroke       0.95      0.99      0.97      3662

    accuracy                           0.97      7324
   macro avg       0.97      0.97      0.97      7324
weighted avg       0.97      0.97      0.97      7324

Accuracy: 0.9706444565811032
AUC: 0.9706444565811032
```

Confusion Matrix

True Negatives (TN)	Correctly predicted "No Stroke" cases: 3486
False Positives (FP)	"No Stroke" cases misclassified as "Stroke": 176
False Negatives (FN)	Stroke" cases misclassified as "No Stroke": 39
True Positives (TP)	Correctly predicted "Stroke" cases: 3623

Overall Model Performance

Accuracy	The model correctly predicted 97% of all stroke cases, indicating strong overall performance.
Macro & Weighted average	Both macro and weighted averages are 0.97, confirming that the model performs consistently well across both classes.
AUC	The AUC of 0.97 indicates that the model has excellent discriminatory ability, distinguishing between the two classes (stroke and no stroke) with a high level of accuracy.

Model 1 Class Metrics Analysis

Class	Precision	Recall	F1-Score
0 (No stroke)	Precision is **0.99**, meaning the model is highly reliable in predicting "No Stroke" cases, with very few false positives	Recall is **0.95**, meaning the model identifies 95% of actual "No Stroke" cases, but misses 5%.	The "No Stroke" class has a 97% rate of balance between recall and precision
1 (Stroke)	Precision is **0.95**, slightly lower, indicating that a small proportion of predicted "Stroke" cases are false positives	The recall is **0.99**, meaning the model identifies 99% of actual "Stroke" cases, with only 1% missed.	The "Stroke" class has a 97% rate of balance between recall and precision
Insights	The model is slightly better at avoiding false positives for "No Stroke" cases than for "Stroke" cases.	The model prioritizes capturing true "Stroke" cases, which is critical in healthcare scenarios where missing a stroke could have severe consequences.	The balanced F1-scores indicate that the model performs equally well for both classes when considering both precision and recall.

3.1.2 Model 2: k-NN with k=2 and metric = Manhattan

As an extra step, I included the Manhattan metric (Boehmke, B., & Greenwell, B. (2019)). It is a distance measure used in k-Nearest Neighbors (k-NN) classification to calculate the distance between two points in a feature space. The Manhattan metric is particularly effective when the features are independent and their differences are meaningful on an absolute scale. Unlike the Euclidean distance, it does not involve squaring, making it less sensitive to outliers.
I performed cross-validation with ten splits, k = 2 and metric = Manhattan, and the accuracy is as follows:

```
[0.95225102, 0.9781719, 0.98090041, 0.98090041, 0.98360656, 0.98360656,
 0.9795082, 0.9795082, 0.97540984, 0.97404372]
Accuracy: 0.98 (+/- 0.02)
```

Below is a summary of the performance metrics:

```
Confusion matrix
[[3539 123]
 [ 47 3615]]

              precision    recall  f1-score   support

No stroke        0.99      0.97      0.98       3662
 Stroke          0.97      0.99      0.98       3662

   accuracy                           0.98       7324
  macro avg        0.98      0.98      0.98       7324
weighted avg       0.98      0.98      0.98       7324

Accuracy: 0.976788640087384
AUC: 0.976788640087384
```

Confusion Matrix

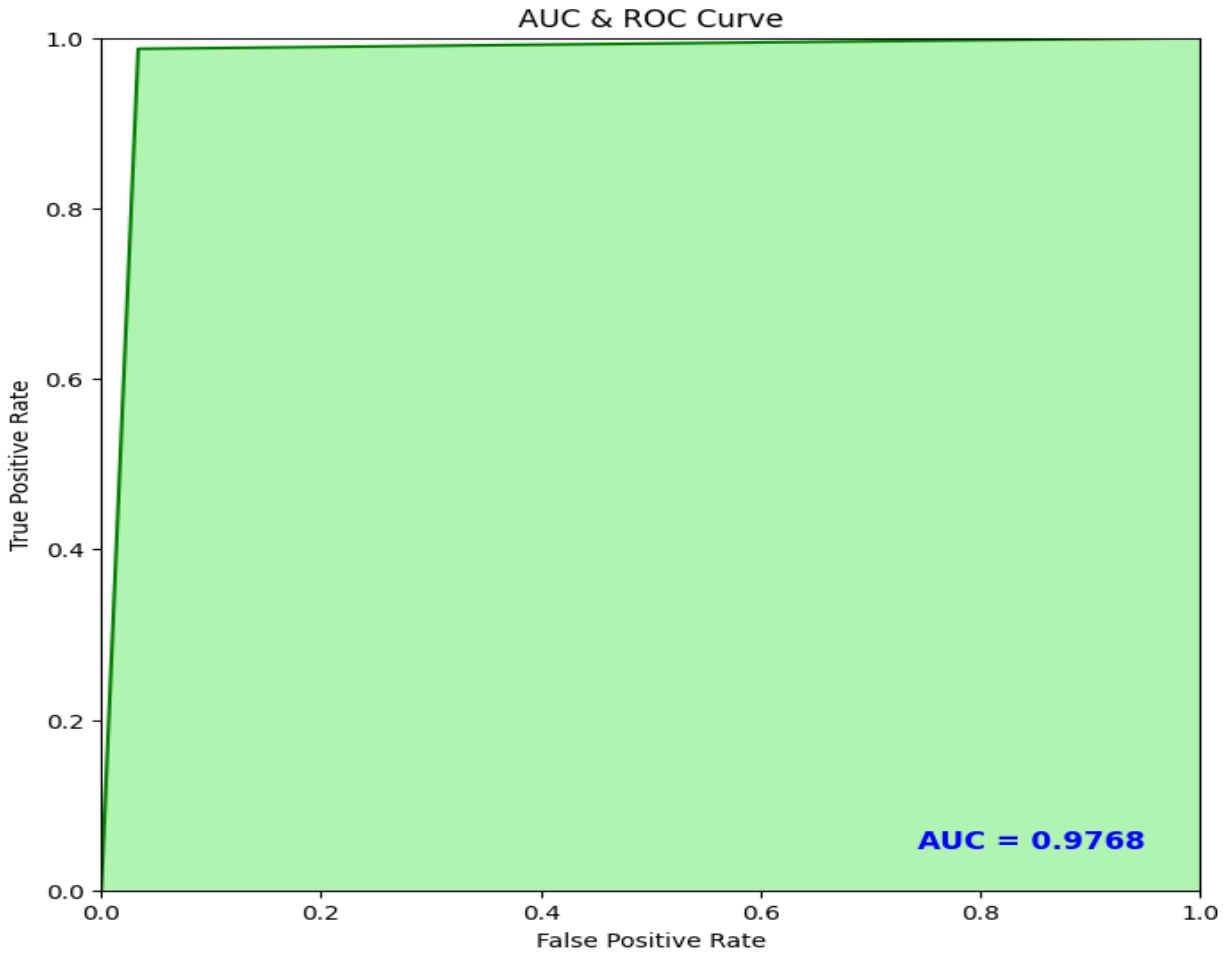
True Negatives (TN)	Correctly predicted "No Stroke" cases: 3539
False Positives (FP)	"No Stroke" cases misclassified as "Stroke": 123
False Negatives (FN)	Stroke" cases misclassified as "No Stroke": 47
True Positives (TP)	Correctly predicted "Stroke" cases: 3615

Overall Model Performance

Accuracy	The model correctly predicted 98% of all stroke cases, indicating strong overall performance.
Macro & Weighted average	Both macro and weighted averages are 0.98, confirming that the model performs consistently well across both classes.
AUC	The AUC of 0.98 indicates that the model has excellent discriminatory ability, distinguishing between the two classes (stroke and no stroke) with a high level of accuracy.

Model 2 Class Metrics Analysis

Class	Precision	Recall	F1-Score
0 (No stroke)	The model is highly accurate in predicting "No Stroke" cases, with only 1% of predictions being false positives.	The model correctly identifies 97% of the actual "No Stroke" cases, with 3% of the actual "No Stroke" cases being missed.	The "No Stroke" class has a 98% rate of balance between recall and precision
1 (Stroke)	The model correctly predicts 97% of the cases it labels as "Stroke," with 3% being false positives.	The model has a very high recall for "Stroke" cases, correctly identifying 99% of the actual stroke cases.	The "Stroke" class has a 98% rate of balance between recall and precision
Insights	The model is very good at avoiding false positives. However, the precision for "Stroke" cases is slightly lower than for "No Stroke" cases, meaning that there are a few more false positive predictions when classifying "Stroke" cases.	The model performs better in identifying "Stroke" cases (high recall of 0.99). The recall for "No Stroke" is also high, but slightly lower than for "Stroke."	The equal F1 scores of 0.98 for both classes suggest that the model is well-balanced in handling both precision and recall. This indicates that the model is not overemphasizing one class over the other, providing a balanced performance across both "No Stroke" and "Stroke" predictions.



AUC of 0.9768 suggests that Model 2 performs very well in distinguishing between the "Stroke" and "No Stroke" classes. The model has a strong ability to correctly classify both positive and negative cases. The ROC curve is close to the top-left corner, indicating that the model is highly effective at identifying stroke cases while minimizing false positives.

Side-by-side comparison

	Accuracy	AUC	Precision	Recall	F1 Score
kNN with k = 2	0.97	0.97	0.97	0.97	0.97
k-NN with k = 2, Metric = Manhattan	0.98	0.98	0.98	0.98	0.98

Conclusion:

Model 2 (k-NN with k = 2, Metric = Manhattan) is the better model overall. It has higher accuracy, AUC, precision for Class 1 (Stroke), recall for Class 0 (No Stroke), and better F1 scores. The use of the Manhattan distance metric seems to slightly improve the model's performance in distinguishing between the classes. Therefore, **Model 2** provides a more robust and reliable prediction for both "Stroke" and "No Stroke" cases.

3.2 NAÏVE BAYES CLASSIFIER

I performed cross-validation with ten splits and the accuracy values were as follows:

```
[0.75716235, 0.7680764, 0.79126876, 0.77216917, 0.78142077, 0.79781421,
 0.80737705, 0.80737705, 0.78961749, 0.80054645]
Accuracy: 0.79 (+/- 0.03)
```

Below is a summary of the performance metrics:

```
Confusion matrix
[[2701 961]
 [ 597 3065]]

                precision    recall  f1-score   support

No stroke          0.82        0.74      0.78       3662
   Stroke          0.76        0.84      0.80       3662

   accuracy                             0.79       7324
  macro avg        0.79        0.79      0.79       7324
weighted avg       0.79        0.79      0.79       7324

Accuracy: 0.7872747132714364
AUC: 0.7872747132714364
```

Confusion Matrix

True Negatives (TN)	Correctly predicted "No Stroke" cases: 2701
False Positives (FP)	"No Stroke" cases misclassified as "Stroke": 961
False Negatives (FN)	Stroke" cases misclassified as "No Stroke": 597
True Positives (TP)	Correctly predicted "Stroke" cases: 3065

Overall Model Performance

Accuracy	The model correctly predicted 79% of all stroke cases, indicating a good overall performance.
Macro & Weighted average	Both macro and weighted averages are 0.79, confirming that the model performs well across both classes.
AUC	The AUC of 0.79 indicates that the model has good discriminatory ability, distinguishing between the two classes (stroke and no stroke) with a moderate level of accuracy.

Naïve Bayes Class Metrics Analysis

Class	Precision	Recall	F1-Score
0 (No stroke)	Of all instances predicted as "No Stroke," 82% are correct.	The model correctly identifies 74% of the actual "No Stroke" cases, with 27% of the actual "No Stroke" cases being missed.	The "No Stroke" class has a 78% rate of balance between recall and precision.
1 (Stroke)	The model correctly predicts 76% of the cases it labels as "Stroke," with 24% being false positives.	The model correctly identifies 84% of the actual "No Stroke" cases, with 16% of the actual "No Stroke" cases being missed.	The "Stroke" class has an 80% rate of balance between recall and precision.
Insights	The model is good at avoiding false positives. The model performs reasonably well in detecting both "No Stroke" and "Stroke" cases. The precision for "Stroke" (0.76) could be improved to reduce false positives	The recall values for both classes indicate a reasonably balanced model, with slightly better performance in recall for "Stroke" (Class 1). This balance suggests the model is effective at identifying true positives for both classes while maintaining a moderate level of false positives and false negatives.	The F1-scores of 0.78 for "No Stroke" and 0.80 for "Stroke" suggest that the model achieves slightly better performance in detecting "Stroke" cases compared to "No Stroke

3.2.1 Naive Bayes Variance Smoothing (1e – 10)

As an extra step, I implemented variance smoothing with Naïve Bayes in an attempt to tune the model. Variance smoothing is a technique used in Gaussian Naive Bayes to handle numerical stability and prevent issues caused by very small variances in the data.
I performed cross-validation with ten splits and variance smoothing(1e-10) and the accuracy values were as follows:

```
[0.75716235, 0.7680764, 0.79126876, 0.77216917, 0.78142077, 0.79781421,
 0.80737705, 0.80737705, 0.78961749, 0.80054645]
Accuracy: 0.79 (+/- 0.03)
```

Below is a summary of the performance metrics:

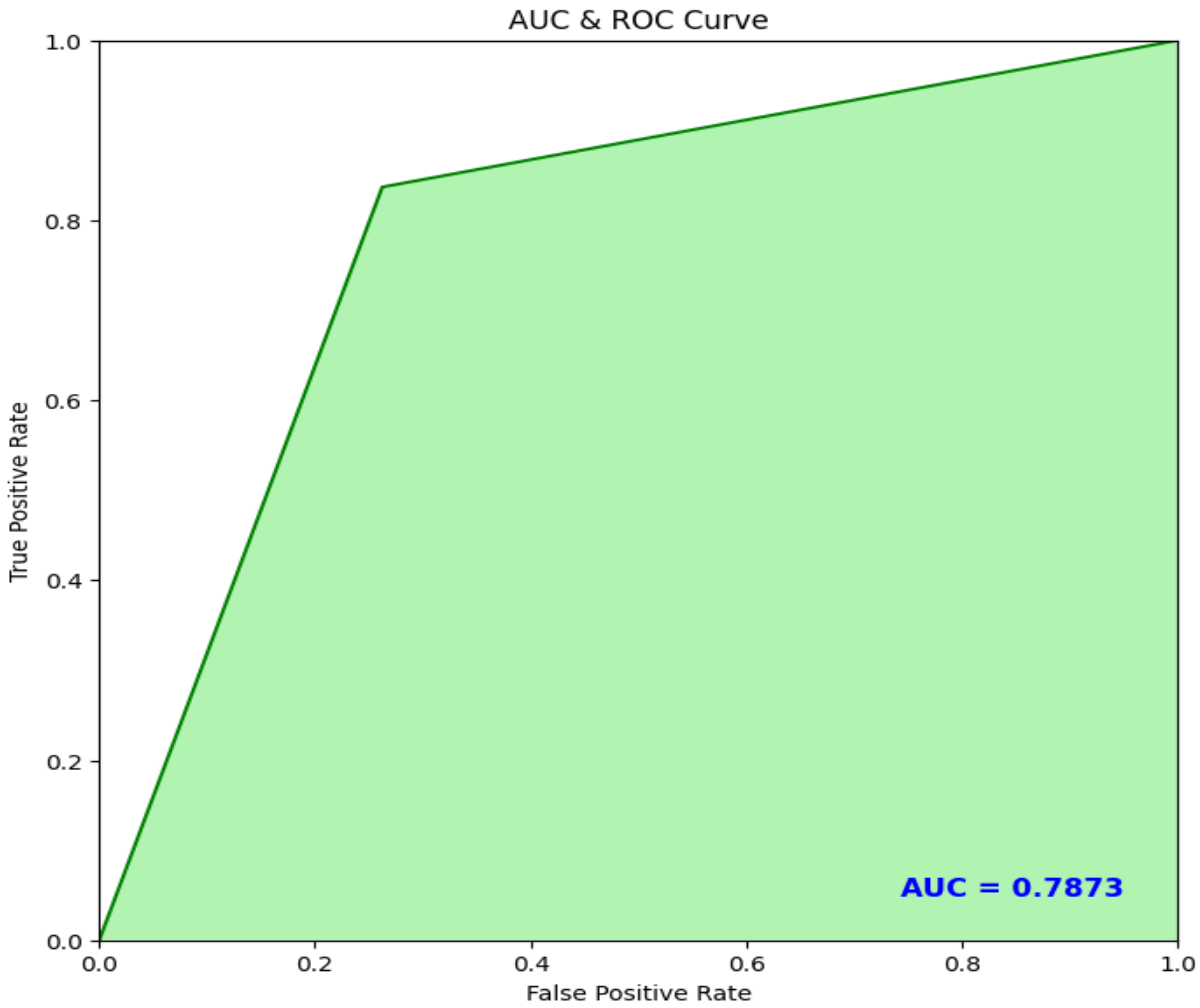
```
Confusion Matrix
[[2701 961]
 [ 597 3065]]

              precision    recall  f1-score   support

No stroke       0.82      0.74      0.78       3662
    Stroke      0.76      0.84      0.80       3662

   accuracy                         0.79       7324
  macro avg     0.79      0.79      0.79       7324
weighted avg    0.79      0.79      0.79       7324

Accuracy: 0.7872747132714364
AUC: 0.7872747132714364
```



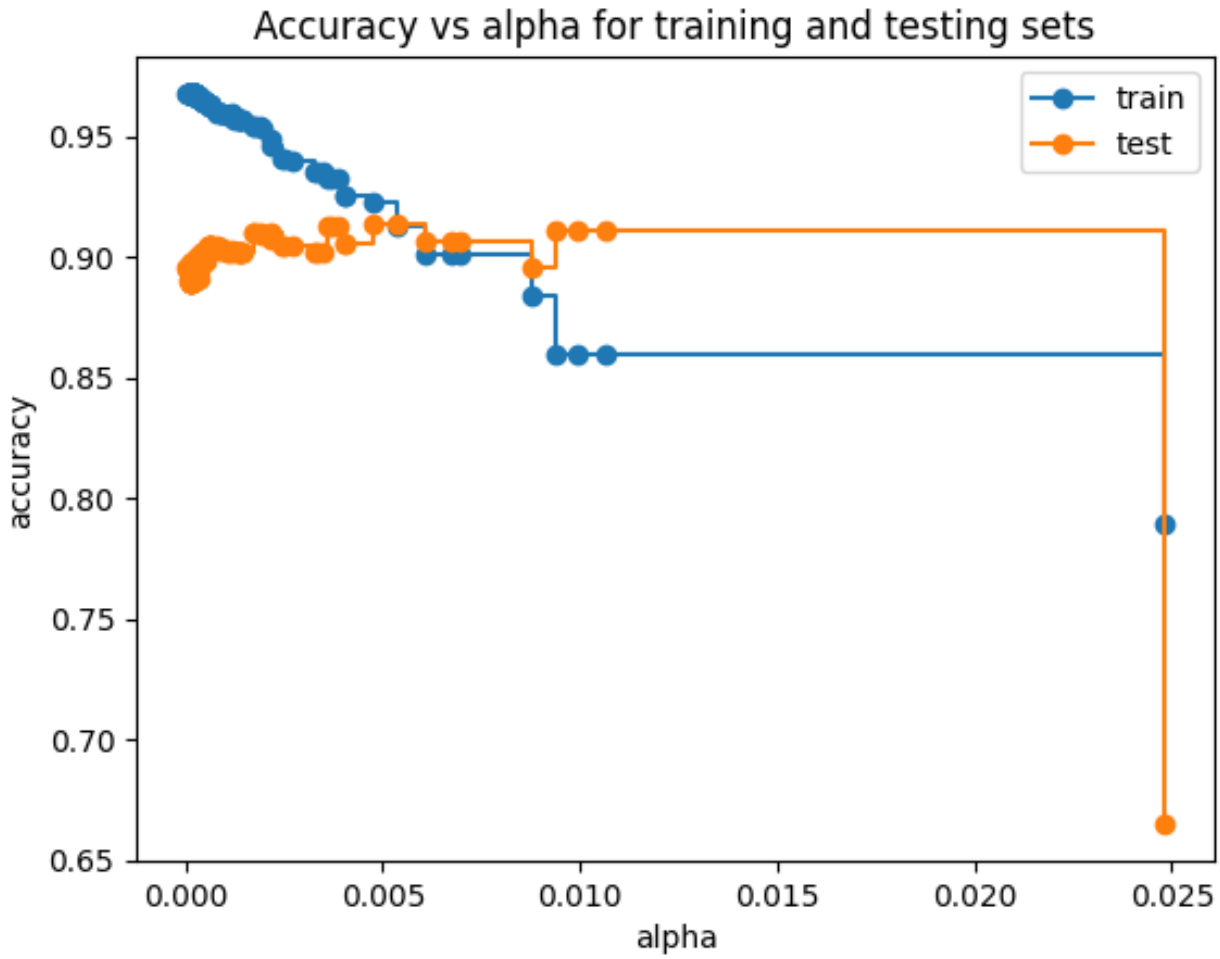
An AUC of 0.787 indicates that the model has a moderate to good ability to distinguish between "Stroke" and "No Stroke" cases. Specifically, there is a 78.7% chance that the model will correctly rank a randomly selected "Stroke" instance higher than a "No Stroke" instance. The model performs better than random guessing, but there is room for improvement, especially in reducing misclassifications.

Conclusion:

Both the models have exactly the same performance meaning that variance smoothing does not have an effect on the model's performance in this case.

3.3 DECISION TREE CLASSIFIER

3.3.1 Decision Tree with criterion='entropy', max_depth = 10, ccp_alpha = 0.005



I explored the effect of the pruning parameter (alpha) on the performance of our decision tree model. The plot above shows the accuracy of the model on both the training and testing sets as alpha varies. The optimal alpha value (0.005) is identified where the testing accuracy is maximized, providing the best balance between model complexity and generalization.

I performed cross-validation with ten splits and (criterion ='entropy', max_depth = 10, ccp_alpha = 0.005) and the accuracy values were as follows:

```
[0.79126876, 0.91814461, 0.95361528, 0.91950887, 0.92622951, 0.91803279,
 0.93579235, 0.91393443, 0.92486339, 0.94262295]
Accuracy: 0.91 (+/- 0.09)
```

Below is a summary of the performance metrics:

```
Confusion matrix
[[3448 214]
 [ 413 3249]]

              precision    recall  f1-score   support

No stroke       0.89          0.94     0.92       3662
    Stroke      0.94          0.89     0.91       3662

   accuracy                           0.91       7324
  macro avg    0.92          0.91     0.91       7324
weighted avg 0.92          0.91     0.91       7324

Accuracy: 0.914391043145822
AUC: 0.914391043145822
```

Confusion Matrix

True Negatives (TN)	Correctly predicted "No Stroke" cases: 3448
False Positives (FP)	"No Stroke" cases misclassified as "Stroke": 214
False Negatives (FN)	Stroke" cases misclassified as "No Stroke": 413
True Positives (TP)	Correctly predicted "Stroke" cases: 3249

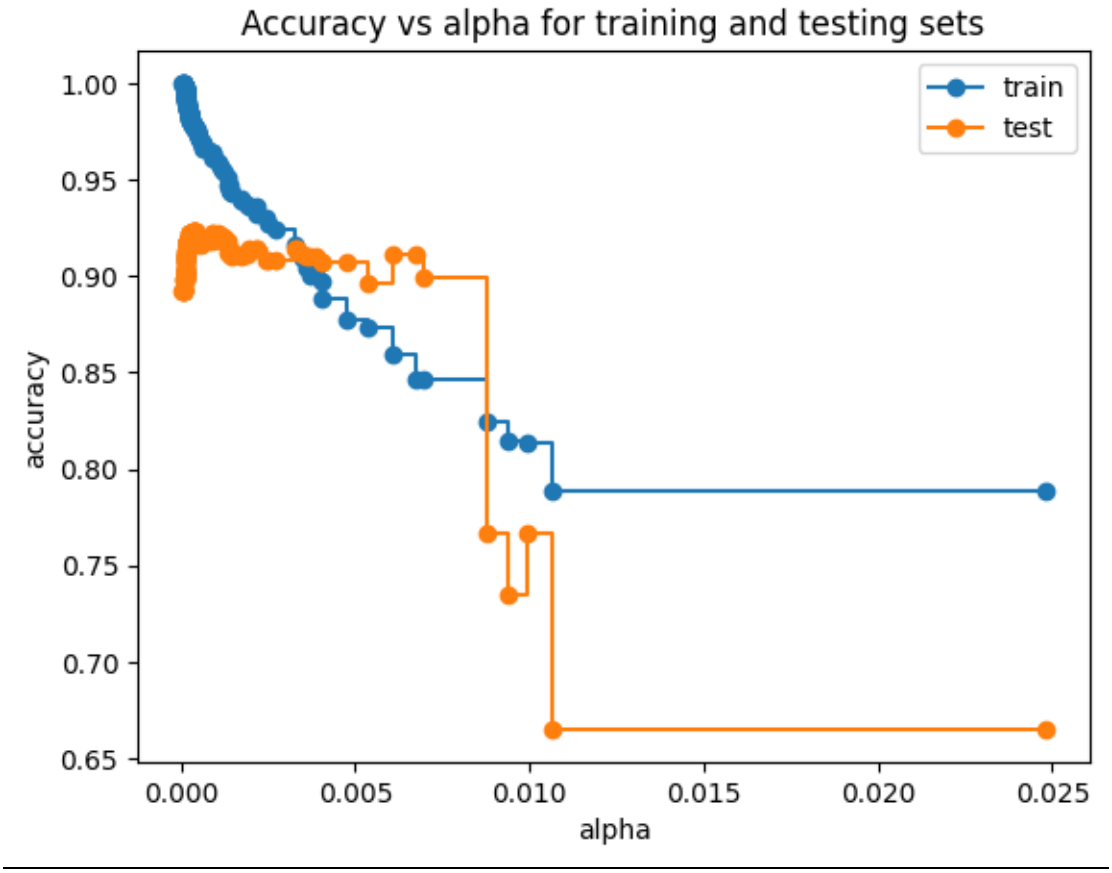
Overall Model Performance

Accuracy	The model correctly predicted 91% of all stroke cases, indicating strong overall performance.
Macro & Weighted average	Both macro and weighted averages are 0.96, confirming that the model performs consistently well across both classes.
AUC	The AUC of 0.91 indicates that the model has great discriminatory ability, distinguishing between the two classes (stroke and no stroke) with a high level of accuracy.

Decision Tree Class Metrics Analysis

Class	Precision	Recall	F1-Score
0 (No stroke)	Of all the instances predicted as "No Stroke," 89% are correct. This indicates that the model is fairly accurate in predicting the "No Stroke" class, with a relatively low false positive rate.	The model correctly identifies 94% of the actual "No Stroke" cases. This is a high recall, indicating that the model is good at detecting the "No Stroke" instances and minimizing false negatives.	The “No Stroke” class has a 92% rate of balance between recall and precision
1 (Stroke)	The model correctly predicts 94% of the cases it labels as "Stroke," with 6% being false positives.	The model correctly identifies 89% of the actual "Stroke" cases. While slightly lower than the recall for "No Stroke," this is still a strong recall, meaning the model is fairly effective at detecting "Stroke" instances.	The “Stroke” class has a 91% rate of balance between recall and precision
Insights	The model is great at predicting "Stroke" cases, with very few false positives.	The model misses a small portion of "Stroke" cases, but it still performs very well in detecting them.	The F1-scores for both classes (0.91) indicate a well-balanced model that performs equally well in detecting both "No Stroke" and "Stroke" cases.

3.3.2 Decision Tree with criterion='entropy', random state = 0, ccp_alpha = 0.001



The plot above shows the model's accuracy on both the training and testing sets as alpha varies. The optimal alpha value (0.001) is identified where the testing accuracy is maximized, providing the best balance between model complexity and generalization

I performed cross-validation with ten splits and (criterion ='entropy', random_state= 0, ccp_alpha = 0.001 and the accuracy values were as follows:

```
[0.78581173, 0.9686221, 0.9781719, 0.9781719, 0.96584699, 0.97404372,
 0.97814208, 0.98224044, 0.97404372, 0.96721311]
Accuracy: 0.96 (+/- 0.11)
```

Below is a summary of the performance metrics:

```
Confusion matrix
[[3556 106]
 [ 222 3440]]

              precision    recall  f1-score   support

    No stroke       0.94      0.97      0.96      3662
       Stroke       0.97      0.94      0.95      3662

     accuracy                           0.96      7324
    macro avg       0.96      0.96      0.96      7324
 weighted avg       0.96      0.96      0.96      7324

Accuracy: 0.955215729109776
AUC: 0.955215729109776
```

Confusion Matrix

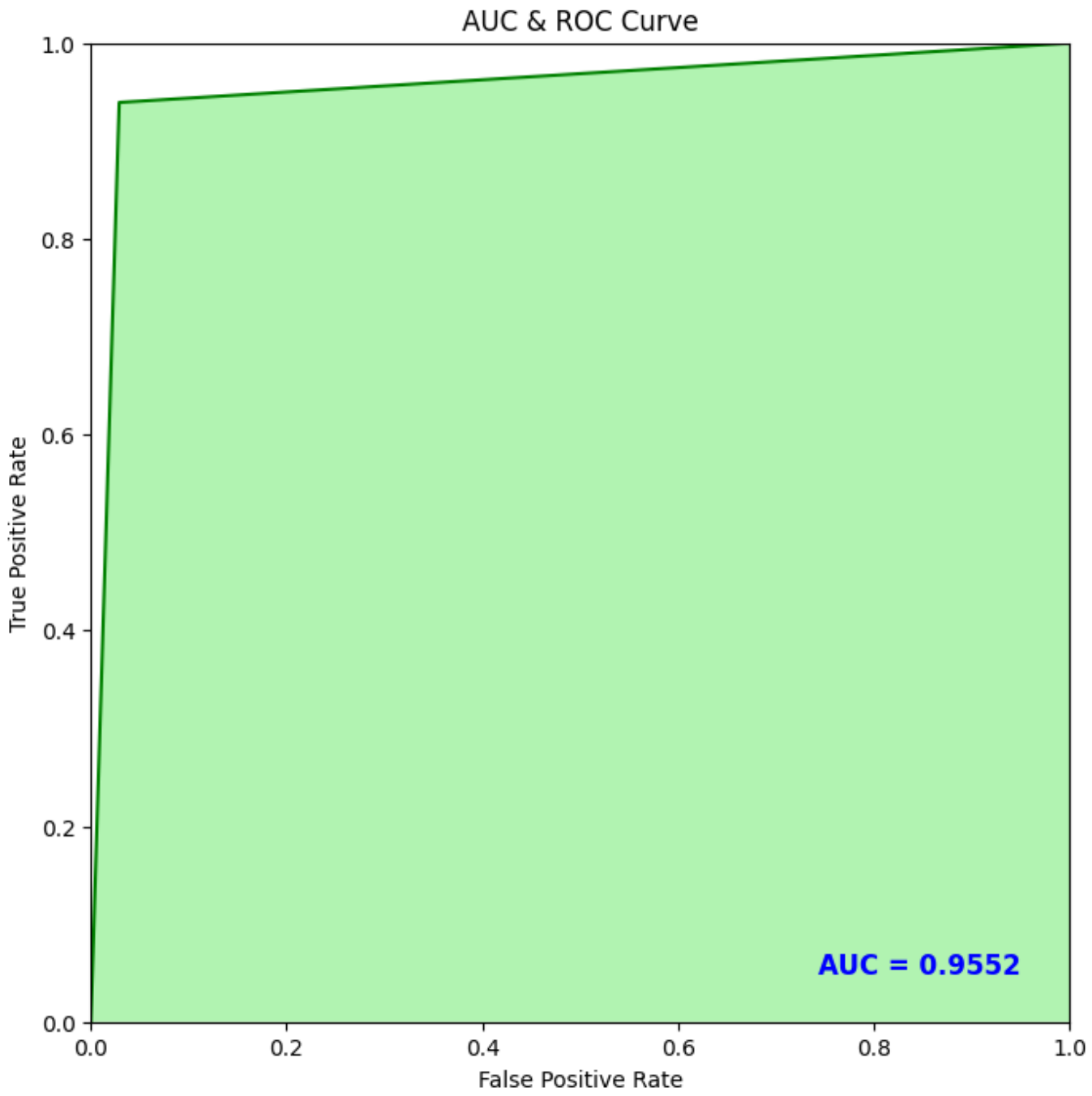
True Negatives (TN)	Correctly predicted "No Stroke" cases: 3556
False Positives (FP)	"No Stroke" cases misclassified as "Stroke": 106
False Negatives (FN)	Stroke" cases misclassified as "No Stroke": 222
True Positives (TP)	Correctly predicted "Stroke" cases: 3440

Overall Model Performance

Accuracy	The model correctly predicted 96% of all stroke cases, indicating strong overall performance.
Macro & Weighted average	Both macro and weighted averages are 0.96, confirming that the model performs consistently well across both classes.
AUC	The AUC of 0.96 indicates that the model has excellent discriminatory ability, distinguishing between the two classes (stroke and no stroke) with a high level of accuracy.

Decision Tree Class Metrics Analysis

Class	Precision	Recall	F1-Score
0 (No stroke)	The model is highly accurate in predicting "No Stroke" cases, with only 6% of predictions being false positives.	The model correctly identifies 97% of the actual "No Stroke" cases, with 3% of the cases being missed.	The "No Stroke" class has a 96% rate of balance between recall and precision
1 (Stroke)	The model correctly predicts 97% of the cases it labels as "Stroke," with 3% being false positives.	The model has a high recall for "Stroke" cases, correctly identifying 94% of the actual stroke cases.	The "Stroke" class has a 95% rate of balance between recall and precision
Insights	The model is excellent at predicting "Stroke" cases, with very few false positives.	The model misses a small portion of "Stroke" cases, but it still performs very well in detecting them.	The F1 scores for both classes suggest that the model is effectively managing both false positives and false negatives, making it reliable for real-world applications, where both precision and recall are important.



AUC of 0.9552 suggests that this model performs very well in distinguishing between the "Stroke" and "No Stroke" classes. The model has a strong ability to correctly classify both positive and negative cases. The ROC curve is close to the top-left corner, indicating that the model is highly effective at identifying stroke cases while minimizing false positives.

Side-by-side comparison

	Accuracy	AUC	Precision	Recall	F1 Score
DT with criterion='entropy', max_depth = 10, ccp_alpha = 0.005	0.91	0.91	0.92	0.91	0.91
DT with criterion='entropy', random state = 0, ccp_alpha = 0.001	0.96	0.96	0.96	0.96	0.96

Conclusion

The decision tree with criterion='entropy', random state = 0, and ccp_alpha = 0.001 is the better performing model overall. It has higher accuracy, better precision, recall, and F1 scores, and lower error metrics. It also has a higher AUC, suggesting superior performance in distinguishing between the classes.

3.4 XGBOOST CLASSIFIER

3.4.1 XGBoost with objective='binary: logistic', random state=42

XGBoost (eXtreme Gradient Boosting) is a powerful and efficient machine learning algorithm based on the gradient boosting framework (Chen, T., & Guestrin, C. (2016)). It builds an ensemble of decision trees sequentially, where each tree corrects the errors of the previous ones. The binary logistic employs the logistic loss function to optimize the model. I performed cross-validation with ten splits and objective='binary: logistic', random state=42, and the accuracy values were as follows:

```
Accuracy scores: [0.78854025, 0.99181446, 0.99181446, 0.9904502, 0.99180328, 0.98907104,
 0.99180328, 0.98907104, 0.99590164, 0.98360656]
```

Below is a summary of the performance metrics:

```
Confusion Matrix:
[[3602   60]
 [ 157 3505]]

Classification Report:
              precision    recall  f1-score   support

No stroke       0.96      0.98      0.97      3662
    Stroke      0.98      0.96      0.97      3662

   accuracy                           0.97      7324
  macro avg     0.97      0.97      0.97      7324
weighted avg    0.97      0.97      0.97      7324

Accuracy: 0.9703713817586018
AUC Score: 0.9881477920908822
```

Confusion Matrix

True Negatives (TN)	Correctly predicted "No Stroke" cases: 3602
False Positives (FP)	"No Stroke" cases misclassified as "Stroke": 60
False Negatives (FN)	Stroke" cases misclassified as "No Stroke": 157
True Positives (TP)	Correctly predicted "Stroke" cases: 3505

Overall Model Performance

Accuracy	The model correctly predicted 97% of all stroke cases, indicating strong overall performance.
Macro & Weighted average	Both macro and weighted averages are 0.97, confirming that the model performs consistently well across both classes.
AUC	The AUC of 0.988 indicates that the model has excellent discriminatory ability, distinguishing between the two classes (stroke and no stroke) with a high level of accuracy.

XGBoost Class Metrics Analysis

Class	Precision	Recall	F1-Score
0 (No stroke)	The model is highly accurate in predicting "No Stroke" cases, with only 4% of predictions being false positives.	The model correctly identifies 98% of the actual "No Stroke" cases, with 2% of the cases being missed.	The "No Stroke" class has a 97% rate of balance between recall and precision
1 (Stroke)	The model correctly predicts 98% of the cases it labels as "Stroke," with 2% being false positives.	The model has a high recall for "Stroke" cases, correctly identifying 96% of the actual stroke cases.	The "Stroke" class has a 97% rate of balance between recall and precision
Insights	The model is excellent at predicting "Stroke" cases, with few false positives.	The model misses a small portion of "Stroke" cases, but it still performs very well in detecting them.	The F1-scores for both classes (0.97) indicate a well-balanced model that performs equally well in detecting both "No Stroke" and "Stroke" cases.

3.4.2 XGBoost with objective = 'binary:logistic', max_depth = 10, random_state = 42

I performed cross-validation with ten splits and objective = 'binary:logistic', max_depth = 10, random_state = 42 and the accuracy values were as follows:

```
Accuracy scores: [0.79263302, 0.9904502, 0.99590723, 0.98499318, 0.9931694, 0.98907104,
 0.99043716, 0.99043716, 0.99180328, 0.98497268]
```

Below is a summary of the performance metrics:

```
Confusion Matrix:
[[3600   62]
 [ 155 3507]]

Classification Report:
              precision    recall  f1-score   support

No stroke       0.96      0.98      0.97      3662
    Stroke      0.98      0.96      0.97      3662

    accuracy                        0.97      7324
   macro avg    0.97      0.97      0.97      7324
weighted avg    0.97      0.97      0.97      7324

Accuracy: 0.9703713817586018
AUC Score: 0.9912006821054113
```

Confusion Matrix

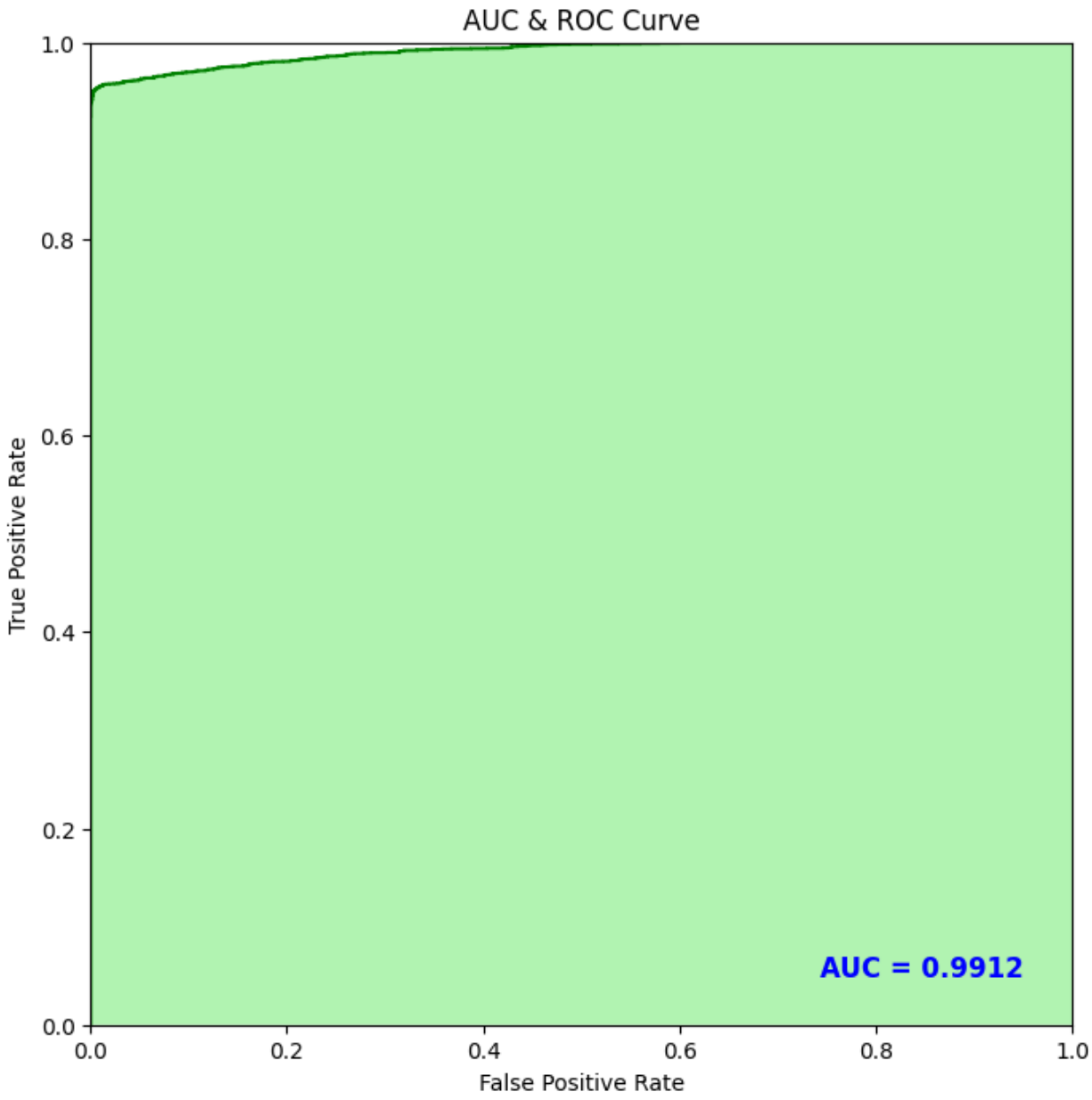
True Negatives (TN)	Correctly predicted "No Stroke" cases: 3600
False Positives (FP)	"No Stroke" cases misclassified as "Stroke": 62
False Negatives (FN)	Stroke" cases misclassified as "No Stroke": 155
True Positives (TP)	Correctly predicted "Stroke" cases: 3507

Overall Model Performance

Accuracy	The model correctly predicted 97% of all stroke cases, indicating strong overall performance.
Macro & Weighted average	Both macro and weighted averages are 0.97, confirming that the model performs consistently well across both classes.
AUC	The AUC of 0.99 indicates that the model has excellent discriminatory ability, distinguishing between the two classes (stroke and no stroke) with a very high level of accuracy.

XGBoost Class Metrics Analysis

Class	Precision	Recall	F1-Score
0 (No stroke)	The model is highly accurate in predicting "No Stroke" cases, with only 4% of predictions being false positives.	The model correctly identifies 98% of the actual "No Stroke" cases, with 2% of the cases being missed.	The "No Stroke" class has a 97% rate of balance between recall and precision
1 (Stroke)	The model correctly predicts 98% of the cases it labels as "Stroke," with 2% being false positives.	The model has a high recall for "Stroke" cases, correctly identifying 96% of the actual stroke cases.	The "Stroke" class has a 97% rate of balance between recall and precision
Insights	The model is excellent at predicting "Stroke" cases, with few false positives.	The model misses a small portion of "Stroke" cases, but it still performs very well in detecting them.	The F1-scores for both classes (0.97) indicate a well-balanced model that performs equally well in detecting both "No Stroke" and "Stroke" cases.



The AUC of 0.9912 suggests that this model performs excellently well in distinguishing between the "Stroke" and "No Stroke" classes. The model is strong in its ability to correctly classify both positive and negative cases. The ROC curve is close to the top-left corner, indicating that the model is highly effective at identifying stroke cases while minimizing false positives.

Side-by-side comparison

	Accuracy	AUC	Precision	Recall	F1 Score
XGBoost with objective='binary: logistic', random state=42	0.97	0.988	0.97	0.97	0.97
XGBoost with objective = 'binary: logistic', max_depth = 10, random state = 42	0.97	0.991	0.97	0.97	0.97

Conclusion

The two models have almost the same performance when it comes to the classification of the stroke attribute. Given the marginally higher AUC score for XGBoost with objective = 'binary: logistic', max_depth = 10, random_state = 42, it is technically the better model. This is important because, in medical applications, it's critical to identify as many stroke cases as possible to ensure timely intervention.

3.5 MODEL COMPARISON

Algorithm	Accuracy	AUC	Precision	Recall	F1 Score
kNN with k=2 and metric = Manhattan	0.98	0.977	0.98	0.98	0.98
Naïve Bayes Classifier	0.79	0.787	0.79	0.79	0.79
Decision Tree with criterion='entropy', random state = 0, ccp_alpha = 0.001	0.96	0.955	0.96	0.96	0.96
XGBoost with objective = 'binary: logistic', max_depth = 10, random state = 42	0.97	0.991	0.97	0.97	0.97

Final Conclusion

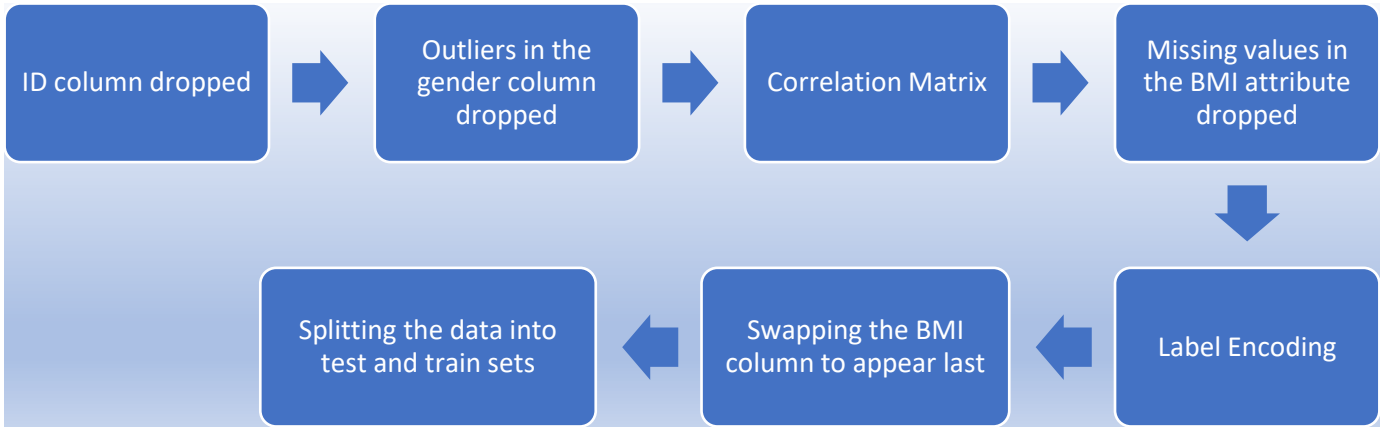
- The **kNN** model achieves the highest accuracy (0.98), precision, recall, and F1 score. However, its AUC (0.977) is slightly lower than XGBoost, indicating slightly less robust discrimination between classes.
- The **Naïve Bayes Classifier** is the weakest model, with significantly lower accuracy (0.79), AUC (0.787), and other metrics. It struggles to classify the stroke attribute effectively.
- **Decision Tree** is a strong performer with an accuracy of 0.96 and an AUC of 0.955. However, kNN and XGBoost slightly overpower it.
- **XGBoost** has an excellent balance of high accuracy (0.97), precision, recall, F1-score, and the highest AUC (0.991), making it the most robust model overall.

• **Best Model**: **XGBoost** is the most robust and reliable model for classifying the stroke attribute due to its high AUC and balanced performance across all metrics.

• **Alternative Model**: If simplicity and interpretability are priorities, **kNN** is a strong alternative, providing comparable performance.

TASK 4: REGRESSION ON THE "BMI" ATTRIBUTE

This is the data preparation pipeline that I followed before performing regression on the BMI attribute.



I used the following regression algorithms: Ordinary Least Squares Regressor and Decision Tree Regressor. The results were as follows:

4.1 ORDINARY LEAST SQUARES REGRESSOR

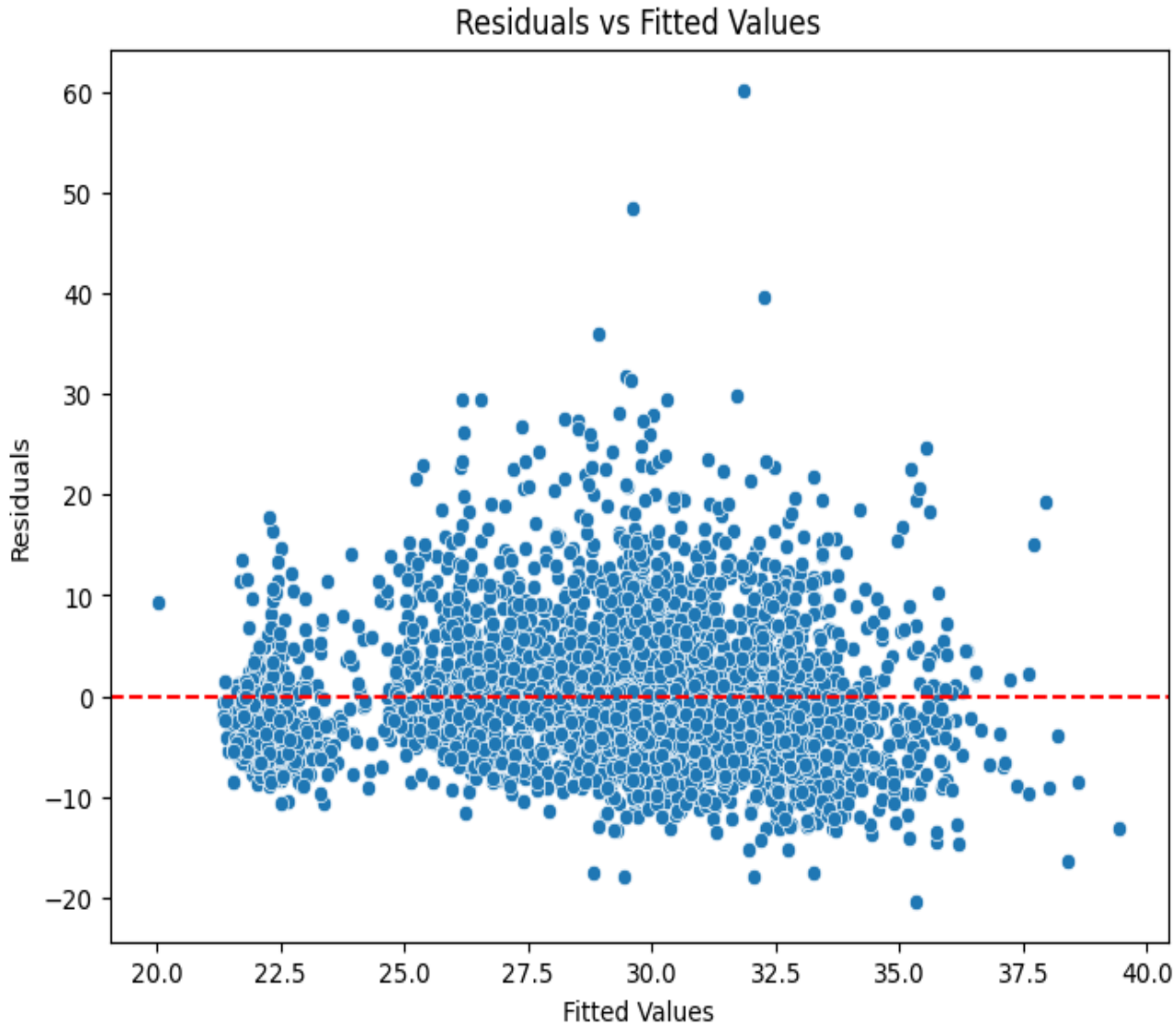
Ordinary Least Squares Regressor (OLS) is a fundamental method in linear regression that estimates the relationship between one or more independent variables (features) and a dependent variable (target). (Seabold, S., & Perktold, J. (2010)) The goal is to find the line that minimizes the sum of squared residuals (differences between observed and predicted values)

Results:

```
Mean Squared Error (MSE): 48.87838543137882
Root Mean Squared Error (RMSE): 6.99130784841998
Mean Absolute Error (MAE): 5.263447682056193
```

```
R-squared:                      0.199
Adj. R-squared:                 0.197
```

Mean Squared Error	The MSE indicates the average squared difference between the predicted and actual BMI values. A value of 48.88 suggests the model's predictions have a moderate error when squared differences are considered.
Root Mean Squared Error	On average, the model's predictions deviate from the true BMI value by 6.99 units.
Mean Absolute Error	On average, the model's prediction of the patient's BMI is off by 5.26 units from the patient's actual BMI.
R-squared	The model explains only 19.9% of the variation in BMI, meaning most factors influencing BMI are not captured by the model.
Adjusted R-squared	After accounting for the number of predictors in the model, it explains only 19.7% of the variation in BMI.



The residuals appear randomly distributed around the red horizontal line at zero, which is a positive sign. This suggests that the model has captured the linear relationship between the predictors and the target variable (BMI) well. There are a few residuals that are significantly far from zero. These represent outliers in the dataset that could impact the model's performance.

4.2 DECISION TREE REGRESSOR

The Decision Tree Regressor predicts the target variable by learning simple decision rules inferred from the data features. The tree is built by recursively splitting the data into subsets based on feature values that minimize the impurity or error at each node. (Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011)).

Results:

```
Mean Absolute Error: 5.081502398673561
Mean Squared Error: 48.081455729652504
Root Mean Squared Error: 6.934079299348436
R2 score: 0.2414000374501849
Adjusted R2: 0.23516155091605817
```

Mean Squared Error	The MSE indicates the average squared difference between the predicted and actual BMI values. A value of 48.08 suggests the model's predictions have a moderate error when squared differences are considered.
Root Mean Squared Error	On average, the model's predictions deviate from the true BMI value by 6.93 units.
Mean Absolute Error	On average, the model's prediction of the patient's BMI is off by 5.08 units from the patient's actual BMI.
R-squared	The model explains only 24.14% of the variation in BMI, meaning most factors influencing BMI are not captured by the model.
Adjusted R-squared	After accounting for the number of predictors in the model, it explains only 23.52% of the variation in BMI.

4.2.1 Predicting Null values in the BMI column using Decision Tree Regressor

As an extra step, I used the Decision Tree Regressor to predict the missing values in the BMI column and these are the steps that I used:

STEP 1: Separate the Rows with Missing and Non-Missing BMI Values	The dataset is split into two subsets: **train_data**: Contains rows where the BMI value is not missing (not-null ()), which will be used for training the model. **test data**: Contains rows where the BMI value is missing (is null ()), which will be used for prediction. This separation is crucial because the model can only be trained on data with known BMI values.
STEP 2: Define Features and Target Variable	**X_train**: This is the feature set used to train the model. It includes all columns from train_data except for BMI. **y_train**: This is the target variable, which in this case is the BMI values from train_data. The model will learn to predict BMI based on the other features. **X_test**: This contains the same features as X_train but for the rows with missing BMI values. These rows will be used to make predictions.
STEP 3: Train the Decision Tree Regressor	A Decision Tree Regressor model is initialized and it is trained using the X_train features and the y_train target variable. The goal is for the model to learn the relationship between the features and the BMI values.
STEP 4: Predict the Missing BMI Values	Once the model is trained, it is used to predict the missing BMI values in X_test. The predictions are stored in an array, which represents the estimated BMI values for the rows that initially had missing data.

The results are as follows:

```
array ([26.4, 30., 27.9, 27.3, 37.5, 34.2, 34.8, 39.2, 40.9, 23.9, 29.,
       30.5, 20.3, 25.3, 35.8, 25.9, 26.1, 34.6, 31.8, 33.7, 26.1, 20.9,
       27., 21.5, 22.6, 27.6, 29.2, 22.1, 28.4, 24.2, 22.9, 29.7, 29.5,
       40.9, 37.9, 22.6, 20.8, 26.1, 18.3, 34.2, 32.5, 31.8, 33.9, 57.7,
       21.9, 34.3, 23.8, 25.8, 45.1, 47.3, 27.5, 33.4, 32.6, 31.1, 27.6,
       20.6, 26.6, 30.2, 24.9, 29.1, 25.9, 29.4, 22.6, 45.2, 40.3, 28.8,
       17.3, 30.7, 32.8, 34.2, 34.8, 36.7, 26.7, 21.9, 24.9, 32.8, 36.4,
       17.4, 45.7, 25.3, 23.2, 35.8, 29.5, 25.1, 21.2, 26.9, 30.8, 27.8,
       34.6, 28.8, 30.7, 22., 37.6, 21.5, 58.1, 32., 57.7, 26.4, 43.8,
       17.3, 23., 20.9, 26.7, 29.5, 38.1, 17.8, 26.1, 36.4, 28.1, 32.8,
       27.6, 28.1, 37.8, 29.5, 42.5, 30.9, 43.1, 33.3, 24.3, 25.6, 32.2,
       32.9, 27.9, 25., 19.9, 40.9, 17.1, 39.6, 23., 38.4, 36.7, 57.5,
       24.9, 24.2, 34.4, 25.8, 19.2, 15., 32.2, 34.4, 34.4, 36.4, 25.8,
       43.4, 56., 32.8, 24.1, 35.3, 31.2, 21.3, 29.3, 34.2, 30.1, 21.7,
       34.2, 42.8, 44.5, 43.9, 42.1, 40.7, 21.4, 18.1, 44.8, 24.9, 28.6,
       33.1, 20.3, 19.7, 40.2, 43.9, 57.5, 33.4, 16.1, 40.4, 28.4, 25.3,
       19.2, 43.4, 57.5, 22.8, 19.3, 31.8, 20.8, 26.2, 32.1, 32.8, 39.1,
       27.3, 25.4, 32.3, 27.4, 22.8, 35.7, 18.9, 30.5, 29.1, 36.9, 27.9,
       43.8, 25.5, 26.9])
```

The Decision Tree Regressor provided a more nuanced way of predicting missing BMI values. This method can account for complex relationships between BMI and other features, potentially leading to more accurate imputed values compared to the median or mean imputation.

4.3 MODEL COMPARISON

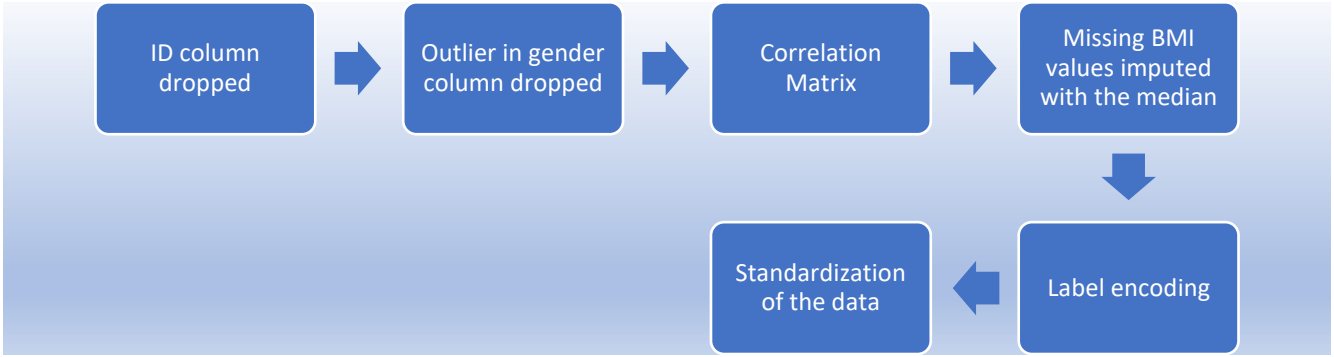
	MSE	RMSE	MAE	R2	ADJ R2
OLS Regressor	48.88	6.99	5.26	0.199	0.197
Decision Tree Regressor	48.08	6.93	5.08	0.241	0.235

Conclusion

The Decision Tree Regressor is slightly better than the OLS Regressor across all metrics, including MSE, RMSE, MAE, R^2, and Adjusted R^2. The Decision Tree model provides better predictive accuracy and explains more variance in the BMI attribute, making it the preferable model for regression in this case.

TASK 5: CLUSTERING

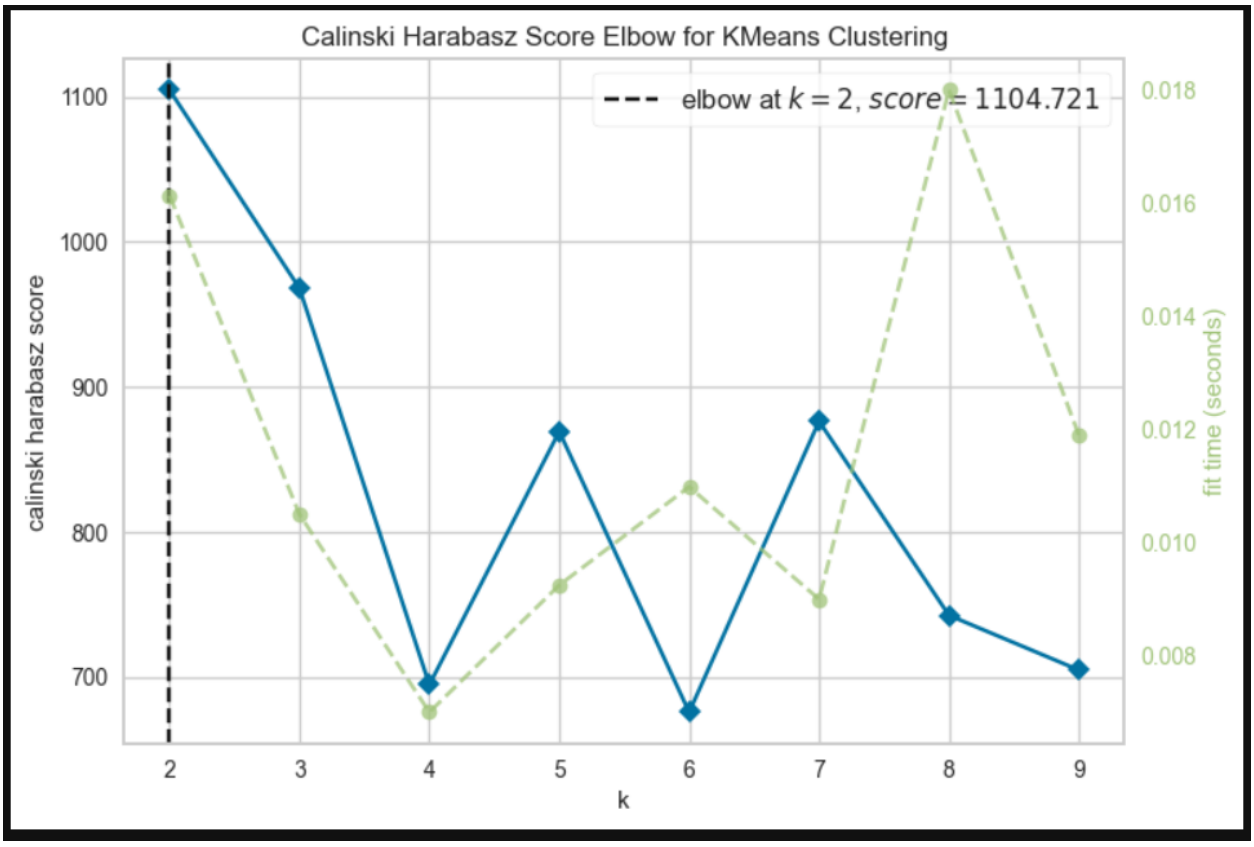
This is the data preparation pipeline I followed before performing clustering on the stroke dataset:



I used two algorithms to perform clustering: **K-Means** algorithm and the **Hierarchical** Clustering algorithm.

5.1: K-MEANS CLUSTERING

I used the Calinski Harabasz score which is used to evaluate the quality of clustering. Since the score for k=2 is the highest (1104.72), this suggests that the clustering solution with 2 clusters is the most optimal. The clusters are likely well-separated and compact compared to other values of k.



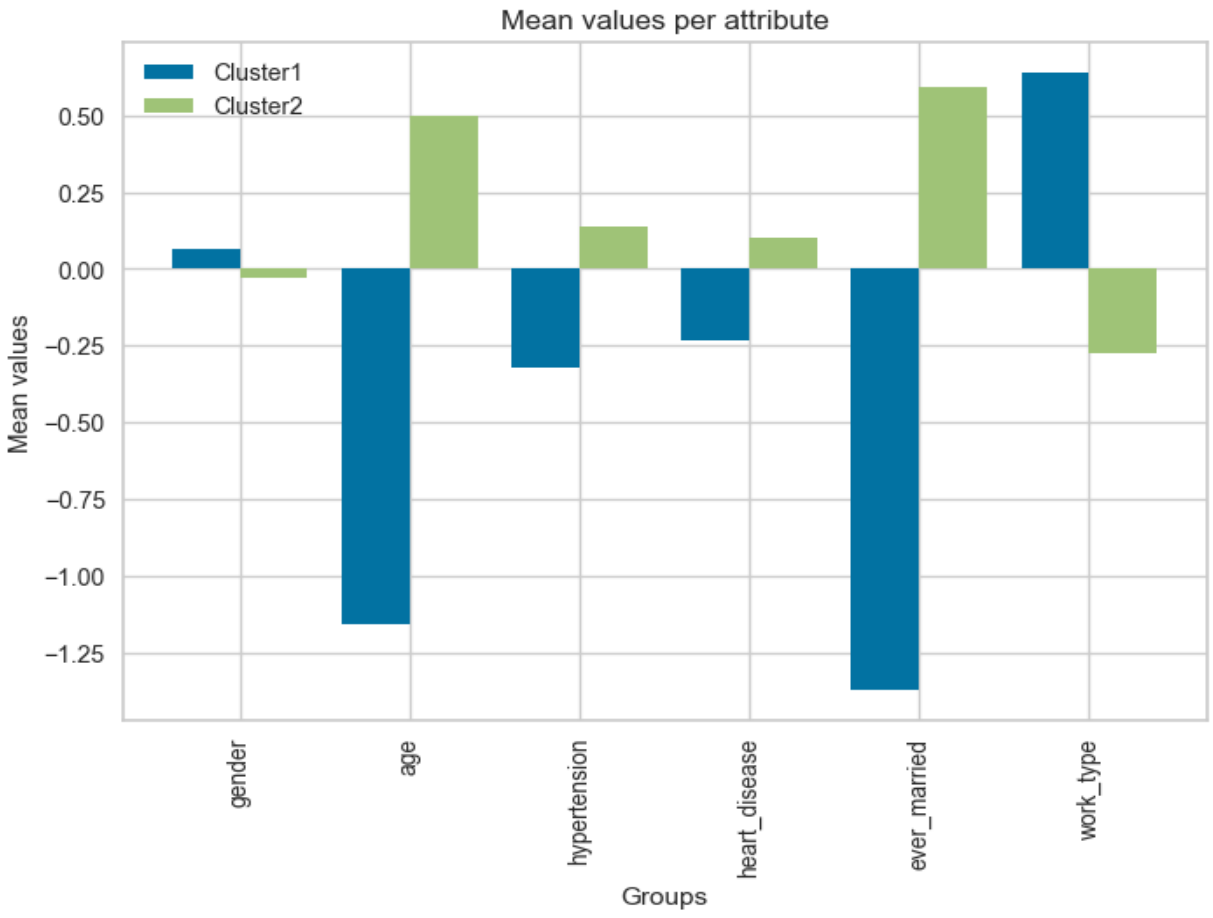
The KMeans model has a **silhouette score of 0.18** meaning that the clustering has some room for improvement, as the points are not very well-clustered with each other.

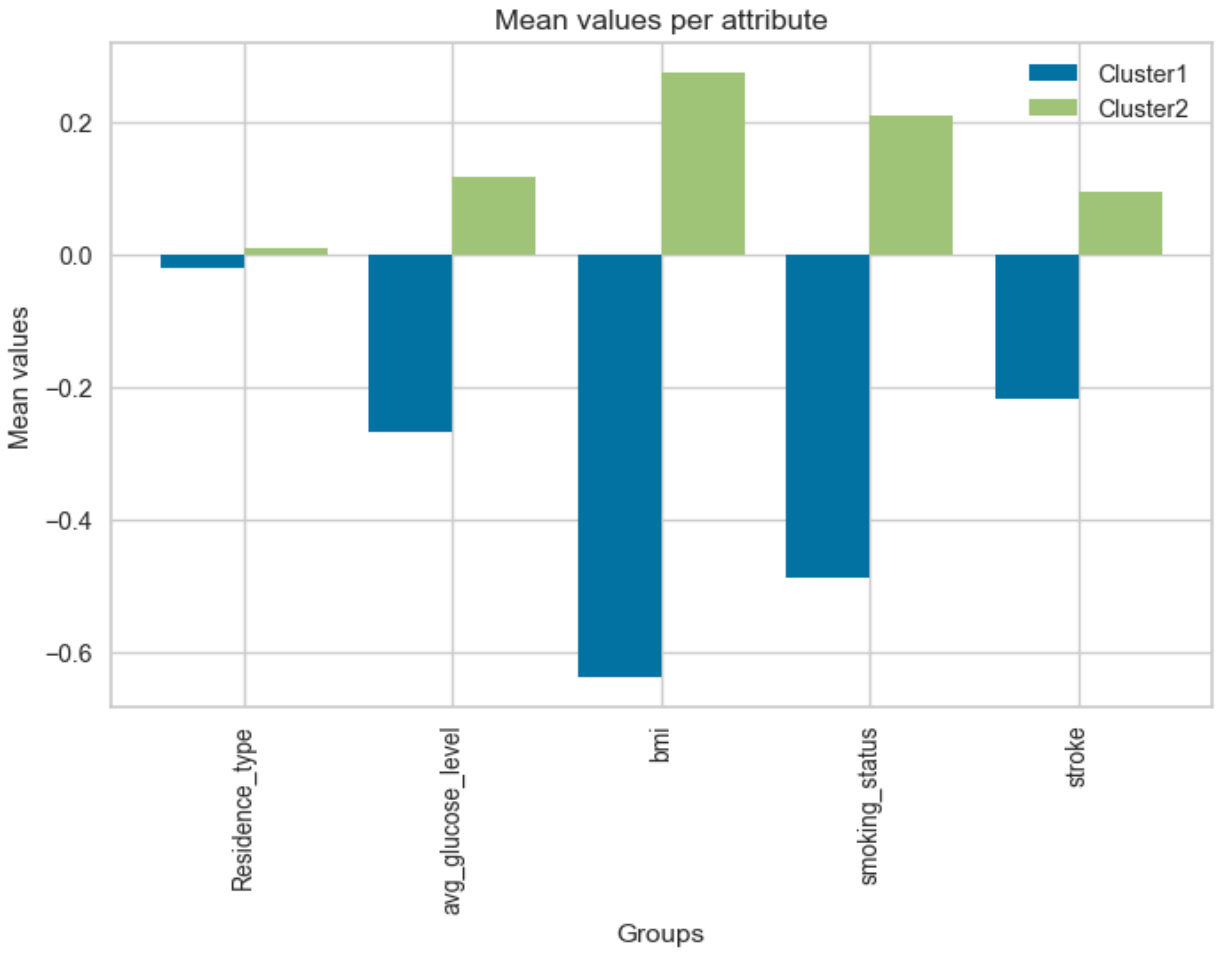
5.1.1 Cluster Shapes

Cluster	Shape	Explanation
1	(1537, 12)	Cluster 1 consists of 1537 data points (rows) and 12 features (columns). The smaller size suggests that it might represent a specific subgroup within the dataset.
2	(3572,12)	Cluster 2 contains 3572 data points (rows) and 12 features (columns). The larger number of data points in this cluster suggests that it represents a more general or diverse group.

5.1.2 Cluster Centers

The cluster centers represent the average value of each attribute for the data points within the clusters. The values are standardized to have a mean of 0 and a standard deviation of 1.




Mean values per attribute
Cluster1
Cluster2
0.2
0.0
−0.2
−0.4
−0.6
Mean values
Residence_type
avg_glucose_level
bmi
smoking_status
stroke
Groups

Cluster 1:

Attribute	Cluster Center	Interpretation
Gender	0.06567669	The gender distribution in this cluster is similar to the overall dataset's mean gender distribution. This suggests a nearly equal or slightly higher proportion of males in this cluster compared to the dataset's average
Age	-1.15566202	The average age of individuals in this cluster is below the overall mean age of the dataset. This cluster primarily contains younger individuals compared to the dataset average.
Hypertension	-0.32205681	Individuals in this cluster are less likely to have hypertension compared to the overall dataset average.
Heart Disease	-0.23321541	Individuals in this cluster are less likely to have heart disease compared to the overall dataset average.
Ever married	-1.36950076	Individuals in this cluster are much less likely to be married compared to the dataset average.
Work type	0.63801959	The work type of individuals in this cluster is above the dataset mean This suggests that individuals in this cluster are more likely to belong to private and government job work types.
Residence Type	-0.02080291	This indicates that the residence type distribution in this cluster is very similar to the overall dataset. This suggests a balanced distribution between rural and urban individuals in this cluster.
Average glucose level	-0.26965414	The average glucose level of individuals in this cluster is below the dataset mean. This suggests that this cluster may represent individuals with relatively lower glucose levels.
BMI	-0.63836978	The average BMI of individuals in this cluster is below the dataset mean. Individuals in this cluster tend to have lower BMI values compared to the overall population.
Smoking status	-0.48783025	Individuals in this cluster are less likely to smoke compared to the dataset average.
Stroke	-0.21728559	Individuals in this cluster are less likely to have experienced a stroke compared to the dataset average. The small magnitude suggests only a slight reduction in stroke prevalence.
Cluster 1 Summary	Cluster 1 primarily represents younger, unmarried patients who exhibit relatively healthy attributes, including lower BMI, glucose levels, and lower prevalence of hypertension, heart disease, and stroke compared to the overall population. Smoking is less common in this cluster, and the work type skews towards higher encoded categories. Gender and residence type distributions are similar to the dataset's average.	

Cluster 2:

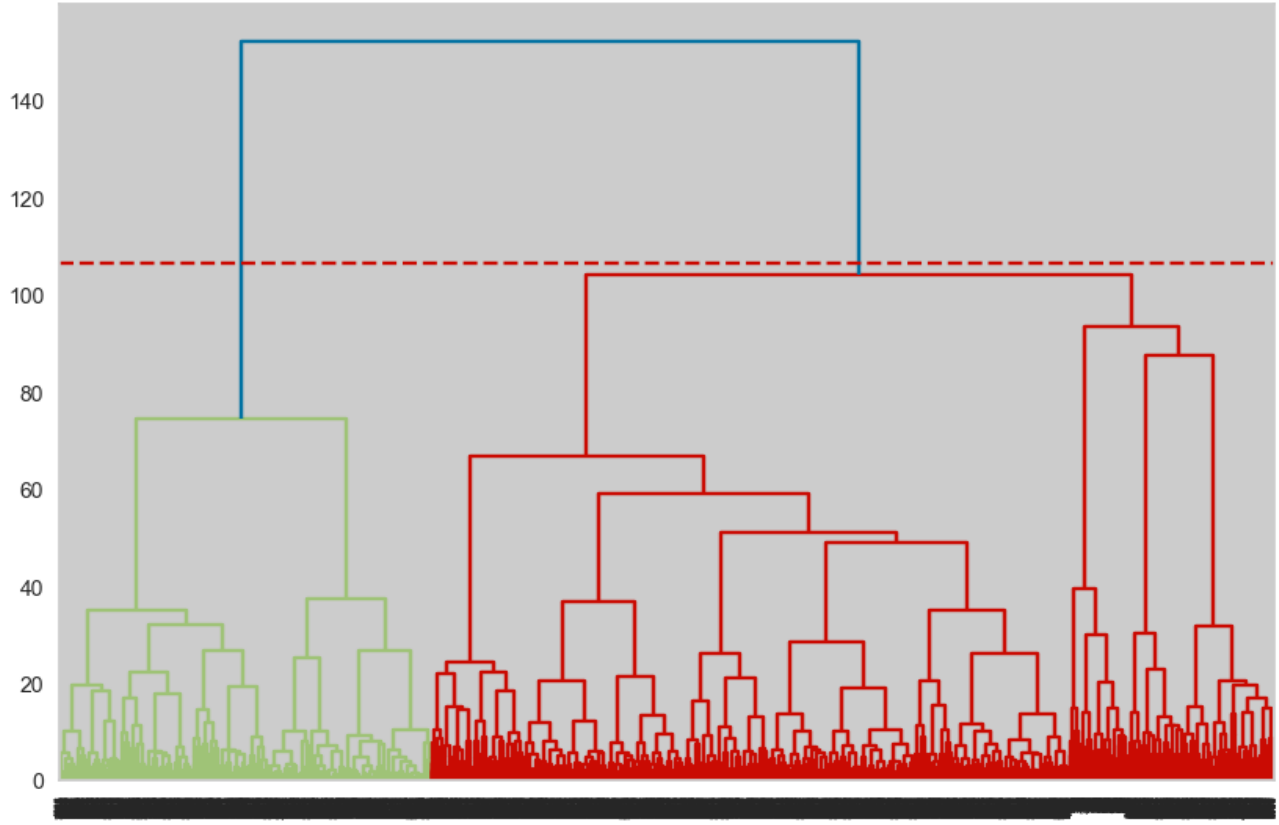
Attribute	Cluster Center	Interpretation
Gender	-0.0282601	The gender distribution in this cluster is very similar to the overall dataset average. This suggests a balanced gender distribution.
Age	0.49727114	Individuals in this cluster are older than the dataset average. The magnitude (0.49) suggests that this cluster represents a moderately older group compared to the dataset mean.
Hypertension	0.1385782	Individuals in this cluster are slightly more likely to have hypertension compared to the dataset average.
Heart Disease	0.10035053	Individuals in this cluster are slightly more likely to have heart disease compared to the dataset average. The magnitude is small, indicating only a minor increase in prevalence.
Ever married	0.58928406	Individuals in this cluster are much more likely to be married compared to the dataset average.
Work type	-0.27453419	The work type of individuals in this cluster is below the dataset mean. This suggests individuals in this cluster are more likely to belong to categories with lower encoded values
Residence Type	0.00895131	The residence type distribution in this cluster is very similar to the overall dataset. This suggests a balanced distribution between rural and urban individuals.
Average glucose level	0.11602979	Individuals in this cluster have slightly higher average glucose levels compared to the dataset mean.
BMI	0.27468487	The average BMI of individuals in this cluster is higher than the dataset's mean. The moderate magnitude suggests that individuals in this cluster tend to have slightly higher BMI values.
Smoking status	0.20990904	Individuals in this cluster are more likely to smoke than the dataset average. The moderate magnitude suggests a noticeable increase in smoking prevalence.
Stroke	0.09349607	Individuals in this cluster are slightly more likely to have experienced a stroke compared to the dataset average. The small magnitude suggests only a minor increase in stroke prevalence.
Cluster 2 Summary	Cluster 2 primarily represents older, married individuals who exhibit a moderately higher prevalence of hypertension, heart disease, and stroke compared to the overall population. This cluster is characterized by slightly higher BMI, glucose levels, and smoking prevalence, indicating less healthy attributes. The work type skews toward lower encoded categories, while the gender and residence type distributions are similar to the dataset's average	

5.1.3 Conclusion

These clusters highlight the contrasting health profiles and demographic characteristics within the dataset, with Cluster 1 reflecting a smaller, healthier subgroup and Cluster 2 representing a larger, less healthy population.

5.2 HIERARCHICAL CLUSTERING

Hierarchical clustering is a method of cluster analysis that seeks to build a hierarchy of clusters. (Köhn, H.-F. (2014)). A dendrogram is a tree-like diagram used to represent the hierarchical structure of data in hierarchical clustering. It visually illustrates how clusters are formed by successively merging or splitting data points based on their similarity or distance.
I plotted a dendrogram to find the optimal number of clusters for this model. From the dendrogram, it seems that the threshold line crosses two main vertical segments. This suggests that the data can be optimally divided into two clusters.

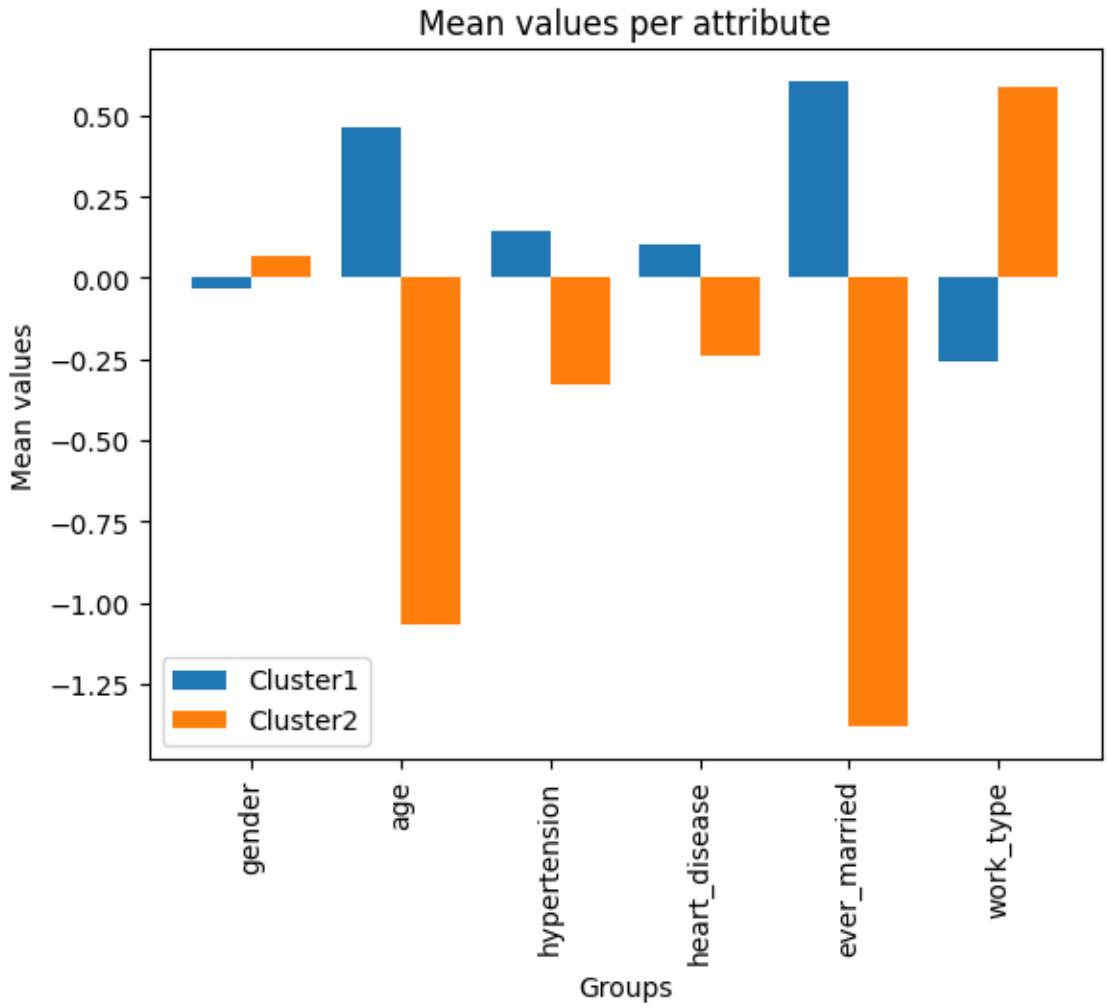


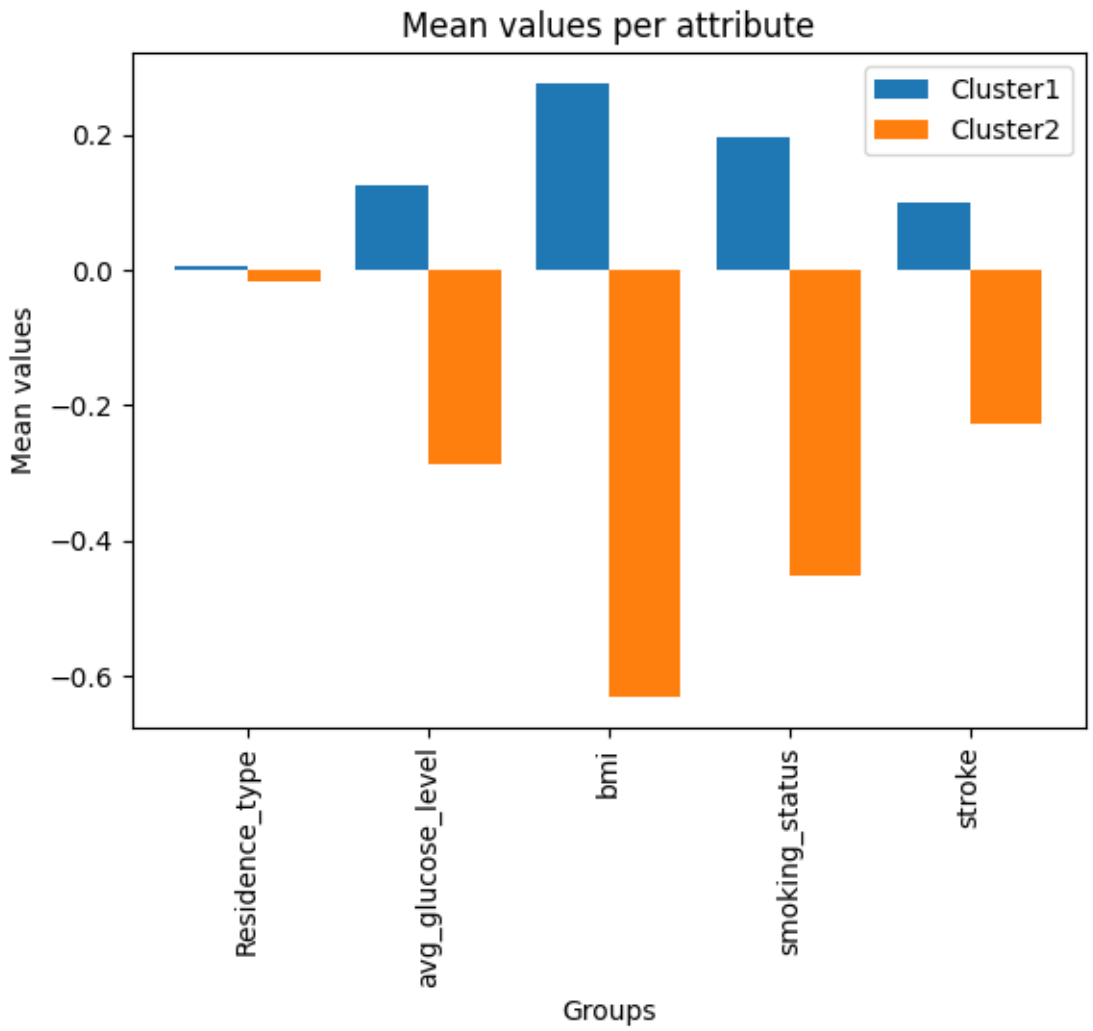
The model has a **silhouette score of 0.17** indicating that the clustering solution has weakly defined clusters, with significant overlap between clusters and less cohesive groupings.

5.2.1 Cluster Shapes

Cluster	Shape	Explanation
1	(3554, 12)	Cluster 1 consists of 3554 data points (rows) and 12 features (columns).
2	(1555,12)	Cluster 2 contains 1555 data points (rows) and 12 features (columns).

5.2.2 Cluster Centers




Mean values per attribute
Cluster1
Cluster2
0.2
0.0
−0.2
−0.4
−0.6
Mean values
Residence_type
avg_glucose_level
bmi
smoking_status
stroke
Groups

Cluster 1:

Attribute	Cluster Center	Interpretation
Gender	-0.03043033	The gender value is slightly below the dataset mean, indicating that the gender distribution in this cluster is close to average and not strongly skewed.
Age	0.46617203	Age in this cluster is significantly above the dataset mean, suggesting that individuals in this cluster are older than the average population in the dataset.
Hypertension	0.14379046	The hypertension value is moderately above the dataset mean, indicating a slightly higher prevalence of hypertension compared to the general dataset.
Heart Disease	0.10455844	Heart disease in this cluster is slightly above the dataset mean, suggesting a marginally higher occurrence of heart disease.
Ever married	0.60459909	This value is substantially above the dataset mean, indicating that individuals in this cluster are much more likely to have been married compared to the average in the dataset.
Work type	-0.25682233	Work type is below the dataset mean, suggesting that individuals in this cluster are less likely to be associated with work types that are more common in the dataset.
Residence Type	0.00682765	The residence type value is very close to the dataset mean, showing no significant difference in urban or rural residence compared to the general dataset.
Average glucose level	0.124911	This value is slightly above the dataset mean, indicating marginally higher average glucose levels in this cluster.
BMI	0.27592401	The BMI value is moderately above the dataset mean, suggesting that individuals in this cluster tend to have higher BMI values than the dataset average.
Smoking status	0.19804887	Smoking status is above the dataset mean, indicating that smoking or related behaviors are more prevalent in this cluster.
Stroke	0.09903632	The stroke value is slightly above the dataset mean, suggesting a marginally higher risk of stroke in this group.
Cluster 1 summary	This cluster is characterized by older individuals, predominantly married, with moderate levels of hypertension, slightly elevated BMI and glucose levels, and a higher prevalence of smoking and stroke risk. Work type and gender differences are less influential in defining this group.	

Cluster 2:

Attribute	Cluster Center	Interpretation
Gender	0.06954945	A slightly positive value suggests a minor skew toward one gender compared to the dataset average. Since the gender is binary (i.e., 0 = female, 1 = male), this cluster has a slightly higher proportion of males.
Age	-1.06545042	This cluster predominantly consists of younger individuals compared to the dataset average.
Hypertension	-0.32863749	Individuals in this cluster are less likely to have hypertension compared to the dataset average.
Heart Disease	-0.23897152	This cluster has a lower heart disease prevalence than the dataset average.
Ever married	-1.38182969	Individuals in this cluster are predominantly unmarried compared to the dataset average.
Work type	0.58697527	The work type for individuals in this cluster skews toward categories with higher encoded values (i.e., private or government jobs).
Residence Type	-0.01560481	A value close to zero suggests that the residence type distribution in this cluster is very similar to the dataset average.
Average glucose level	-0.2854879	Individuals in this cluster have lower average glucose levels, suggesting better metabolic health than the dataset average.
BMI	-0.63063276	Individuals in this cluster have lower BMI values, indicating they are likely to have healthier weight profiles compared to the dataset average.
Smoking status	-0.45264673	Individuals in this cluster are less likely to smoke than the dataset average.
Stroke	-0.22635054	Individuals in this cluster are less likely to have experienced a stroke compared to the dataset average.
Cluster 2 Summary	Cluster 2 is characterized by a younger population with better overall health indicators compared to the dataset average. This group has a lower prevalence of hypertension, heart disease, and stroke, as well as lower BMI, glucose levels, and smoking rates, suggesting a population with a healthier lifestyle and lower risk of chronic conditions. The cluster is predominantly unmarried and skews toward individuals in private or government jobs, while the distribution of residence type is similar to the dataset average. These attributes indicate a relatively low-risk population in terms of health outcomes.	

5.2.3 Conclusion

The clustering analysis identified two distinct groups. **Cluster 1 (3554 individuals)** is characterized by older individuals with higher rates of hypertension, heart disease, and stroke, indicating a high-risk population. **Cluster 2 (1555 individuals)** consists of younger, healthier individuals with lower rates of chronic conditions, representing a low-risk group. These clusters highlight varying health risks, enabling targeted health interventions.

5.3 MODEL COMPARISON

Metric	Values	Interpretation
Silhouette score	***KMeans:*** 0.18 ***Hierarchical:*** 0.17	Both models have relatively low silhouette scores, suggesting that the clusters are not well-separated or distinct. However, KMeans Clustering has a slightly higher silhouette score (0.18), indicating that its clusters may be marginally better defined than those produced by Hierarchical clustering.
Cluster sizes	***KMeans:*** Cluster 1: 1,537 instances Cluster 2: 3,572 instances ***Hierarchical:*** Cluster 1: 3554 instances Cluster 2: 1,555 instances	The two methods differ in how they allocate the majority of the data to one cluster, with KMeans favoring Cluster 2 and Hierarchical favoring Cluster 1.

References

1. Köhn, H.-F. (2014). Hierarchical Cluster Analysis. In Wiley StatsRef: Statistics Reference Online. Wiley. https://doi.org/10.1002/9781118445112.stat02449.pub2

2. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

3. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321–357. https://doi.org/10.1613/jair.953

4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

5. Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and Statistical Modeling with Python.

6. Zafeirelli, S., & Kavroudakis, D. (2024). Comparison of outlier detection approaches in a Smart Cities sensor data context. International Journal on Smart Sensing & Intelligent Systems, 17(1), 1–18. https://doi.org/10.2478/ijssis-2024-0004

7. Boehmke, B., & Greenwell, B. (2019). Hands-On Machine Learning with R. Chapman and Hall/CRC.