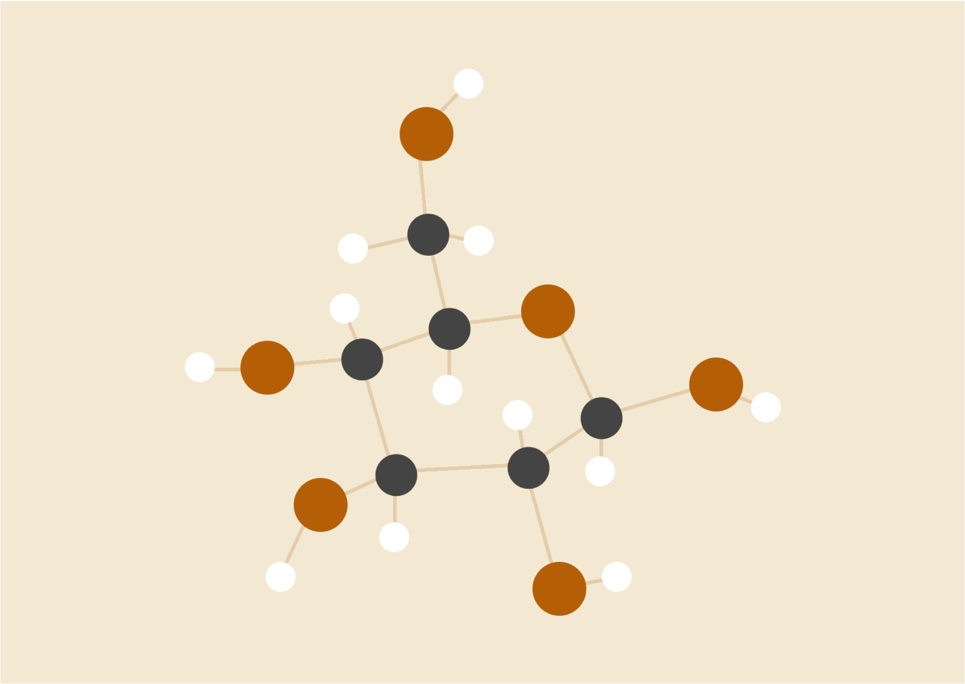
Stroke Prediction Report

Graph Analytics and Algorithms

18MAT333

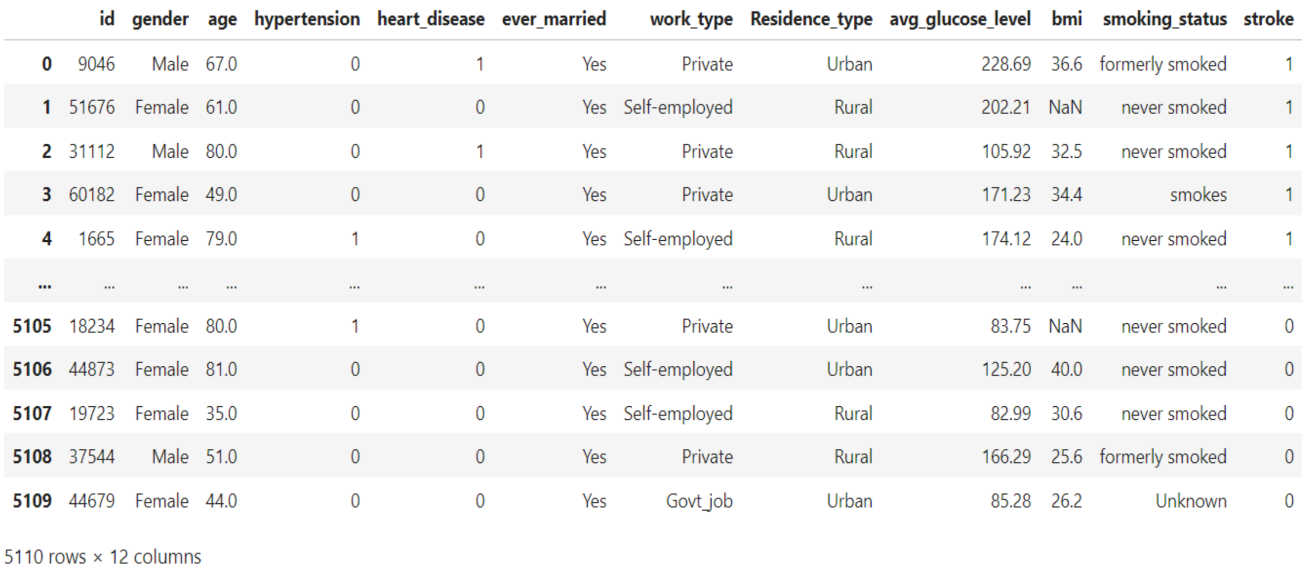


Pratiksha Cauvery K.P - CB.SC.I5DAS21049

Harshitha.T - CB.SC.I5DAS21079

INTRODUCTION

Stroke is a significant global health concern and a leading cause of mortality and disability worldwide. Early detection and prediction of stroke risk factors are crucial for preventive healthcare interventions. In this report, we present an analysis of a stroke prediction dataset aimed at developing machine learning models to predict the likelihood of stroke occurrence based on various demographic, lifestyle, and health-related factors.



	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
...	...	...	...	...	...	...	...	...	...	...	...	...
5105	18234	Female	80.0	1	0	Yes	Private	Urban	83.75	NaN	never smoked	0
5106	44873	Female	81.0	0	0	Yes	Self-employed	Urban	125.20	40.0	never smoked	0
5107	19723	Female	35.0	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked	0
5108	37544	Male	51.0	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked	0
5109	44679	Female	44.0	0	0	Yes	Govt_job	Urban	85.28	26.2	Unknown	0

5110 rows × 12 columns

ABSTRACT

Stroke is a major global health issue, contributing significantly to mortality and disability rates worldwide. Timely identification and prediction of stroke risk factors are essential for effective preventive healthcare measures. In this study, we analyze a heart stroke dataset, aiming to develop machine learning models capable of predicting the likelihood of stroke occurrence based on diverse demographic, lifestyle, and health-related features. Leveraging various preprocessing techniques, graph centrality measures, and advanced machine learning algorithms, including Graph Convolutional Neural Networks (GCNs), we explore the predictive power of different models in identifying stroke risk factors. Our findings provide valuable insights into the effectiveness of these models and their potential applications in early stroke detection and preventive interventions.

ABSTRACT

The objective of this study is to develop and evaluate machine learning models for predicting the likelihood of stroke occurrence based on demographic, lifestyle, and health-related factors. Leveraging a

heart stroke dataset, our analysis focuses on:

1) Implementing data preprocessing techniques to ensure data quality and suitability for machine learning analysis.
2) Assessing the importance of features using graph centrality measures to identify key variables influencing stroke occurrence.
3) Implementing multiple machine learning algorithms, including logistic regression, random forest, and Graph Convolutional Neural Networks (GCNs), to predict stroke risk factors.
4) Analyzing and comparing the performance of different machine learning models in terms of accuracy, precision, recall, and F1-score.
5) Discussing the implications of our findings and providing insights into the potential applications of these models in early stroke detection and preventive healthcare interventions.

DATA OVERVIEW

Data Size :

5110 rows × 12 columns.

Features:

1. **id:** This feature represents a unique identifier for each individual patient in the dataset. It's typically used for indexing and referencing purposes.
2. **gender**: Gender of the patient, categorized as either Male, Female, or Other. It's a categorical feature indicating the gender identity of the individual.
3. **age**: Age of the patient in years. It's a numerical feature representing the age of the individual at the time of data collection.
4. **hypertension**: Indicates whether the patient has hypertension (high blood pressure) or not. It's a binary feature with values 0 (no hypertension) or 1 (hypertension present).
5. **heart_disease**: Indicates whether the patient has any heart disease or not. Similar to hypertension, it's a binary feature with values 0 (no heart disease) or 1 (heart disease present).
6. **ever_married**: Indicates whether the patient has ever been married or not. It's a categorical feature with values Yes or No.

7. **work_type**: Represents the type of work the patient is engaged in. It's a categorical feature with multiple categories such as Private, Self-employed, and Govt_job.
8. **Residence_type**: Indicates the type of residence of the patient, whether Urban or Rural. It's a categorical feature representing the living environment of the individual.
9. **avg_glucose_level**: Average glucose level in the blood of the patient measured in mg/dL (milligrams per deciliter). It's a numerical feature providing information about the patient's blood sugar level.
10. **bmi:** Body Mass Index (BMI) of the patient, calculated as the weight in kilograms divided by the square of the height in meters. It's a numerical feature representing the body composition and weight status of the individual.
11. **smoking_status:** Indicates the smoking status of the patient, categorized as never smoked, formerly smoked, smokes, or Unknown. It's a categorical feature providing information about the smoking habits of the individual.
12. **stroke**: The target variable indicates whether the patient had a stroke event or not. It's a binary feature with values 0 (no stroke) or 1 (stroke present).

Missing Values:

There is one variable that contains missing values. 'bmi' accounts for 3.37% of overall samples. To handle these missing values in 'bmi' variable, we will replace them with the average of 'bmi' values.

Data shape: (5110, 12) before removal of missing values.

Data shape: (3566, 11) after removal of missing values.

Unique Values:

Unique 'gender': ['Male' 'Female' 'Other']

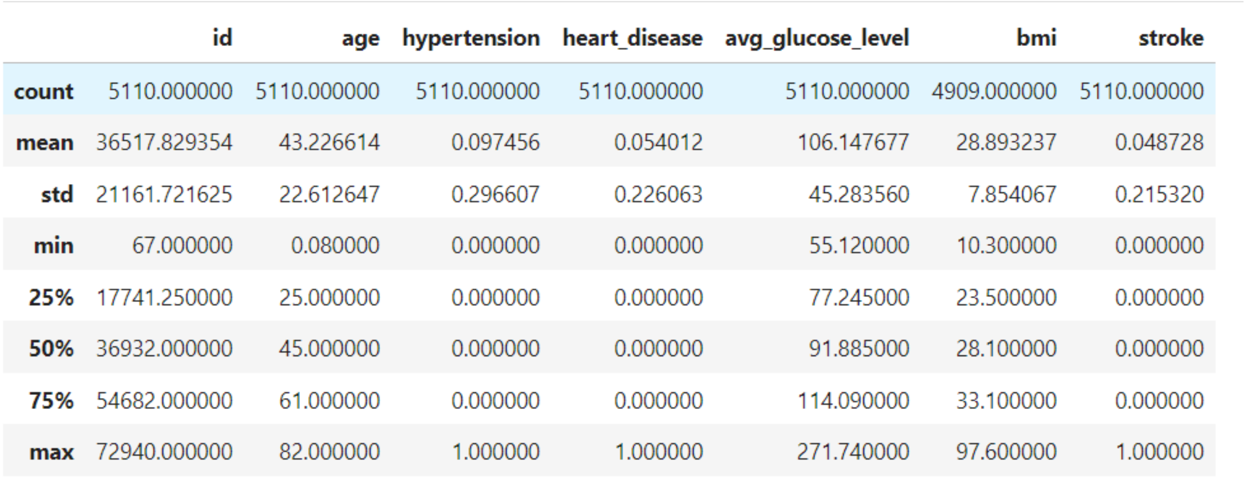
Unique 'ever_married': ['Yes' 'No']

Unique 'work_type': ['Private' 'Self-employed' 'Govt_job' 'children' 'Never_worked']

Unique 'Residence_type': ['Urban' 'Rural']

Unique 'smoking_status': ['formerly smoked' 'never smoked' 'smokes']

Statistical Description:



	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

summary of statistics for each numerical column, including count, mean, standard deviation, minimum, quartiles, and maximum values.

DATA PREPROCESSING

Label Encoding

It is a preprocessing technique used to convert categorical variables into numerical representations, which are essential for many machine learning algorithms. In this analysis, we employed the LabelEncoder class from the sklearn.preprocessing module to perform label encoding on selected categorical features within the dataset. We applied label encoding sequentially to categorical features such as 'gender', 'ever_married', 'work_type', 'Residence_type', and 'smoking_status'.

Gender: The 'gender' column has been encoded into numerical values. 'Male' might have been encoded as 0 and 'Female' as 1.

Ever Married: The 'ever_married' column has been encoded, with 'Yes' possibly represented as 1 and 'No' as 0.

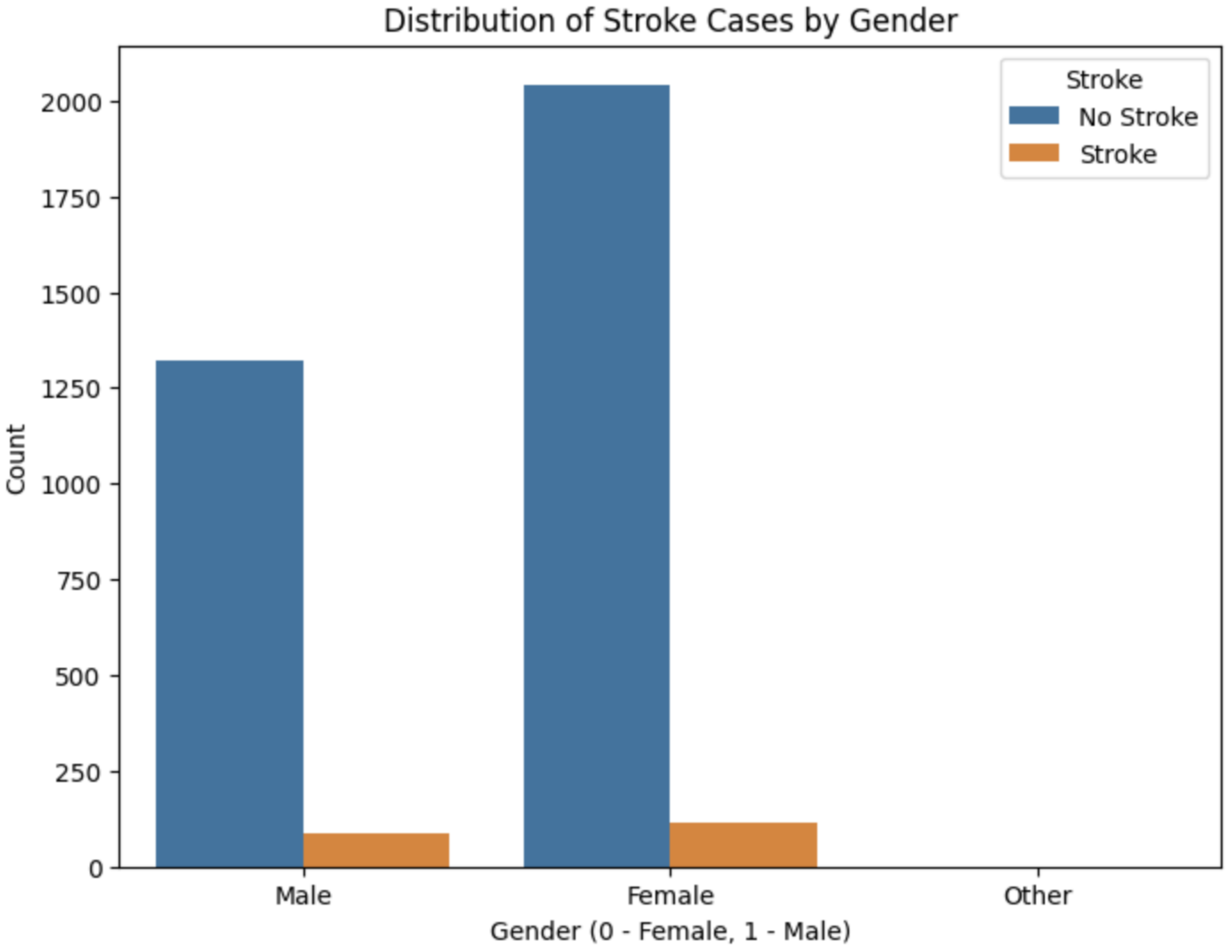
Work Type: The 'work_type' column has also been encoded, with each category assigned a numerical label. 'Private' is represented as 2, 'Self-employed' as 3, and so on.

Residence Type: Similarly, the 'Residence_type' column has been encoded, with 'Urban' and 'Rural' assigned numerical labels, such as 1 and 0.

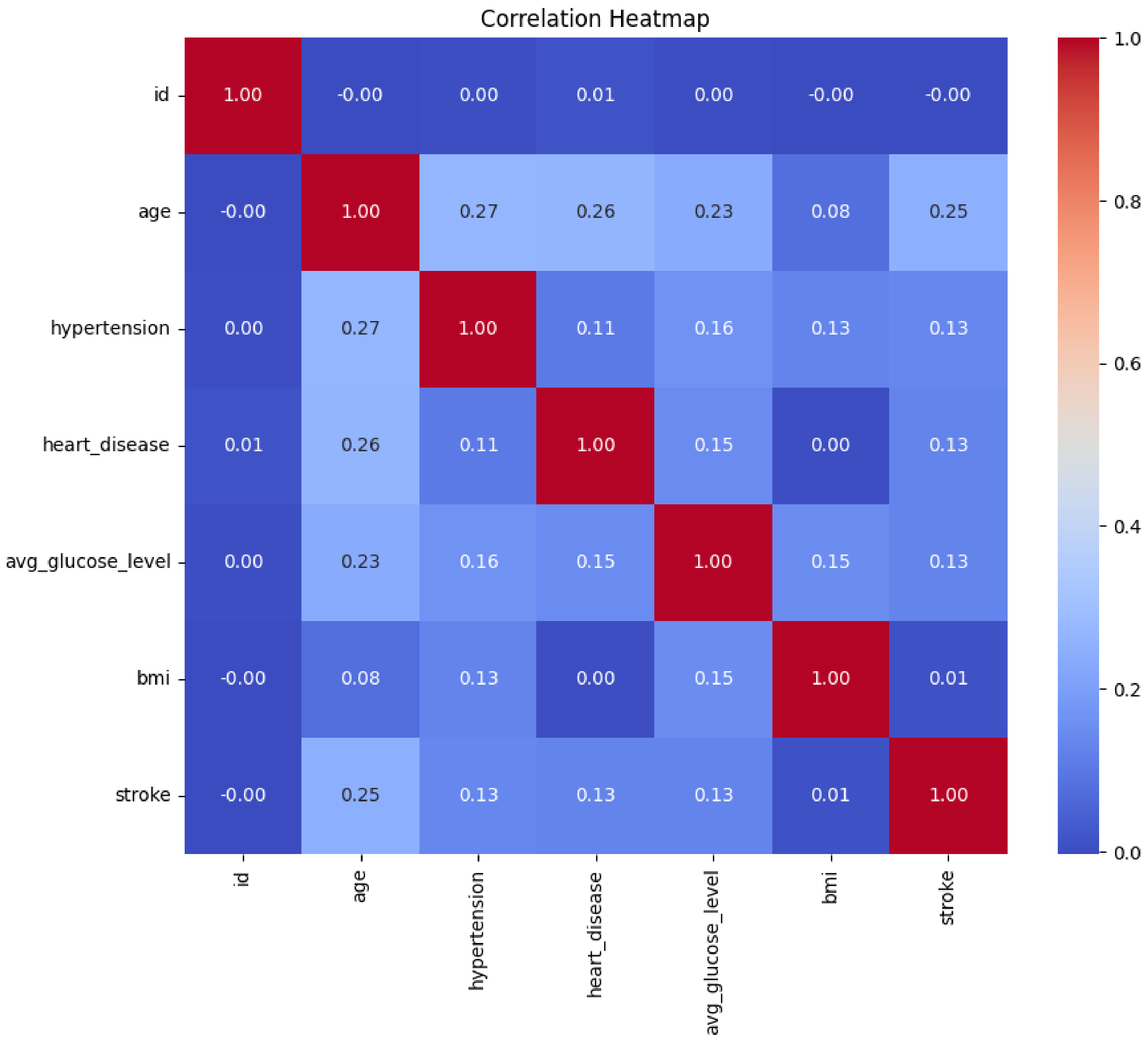
Smoking Status: The 'smoking_status' column has been encoded into numerical values, where different smoking statuses (e.g., 'never smoked', 'formerly smoked', 'smokes') are represented by corresponding integer labels.

Stroke: It appears that the 'stroke' column, which likely represents the target variable, has also been encoded. For binary classification tasks, one class (e.g., 'stroke present') might be encoded as 1, while the other class (e.g., 'no stroke') could be encoded as 0.

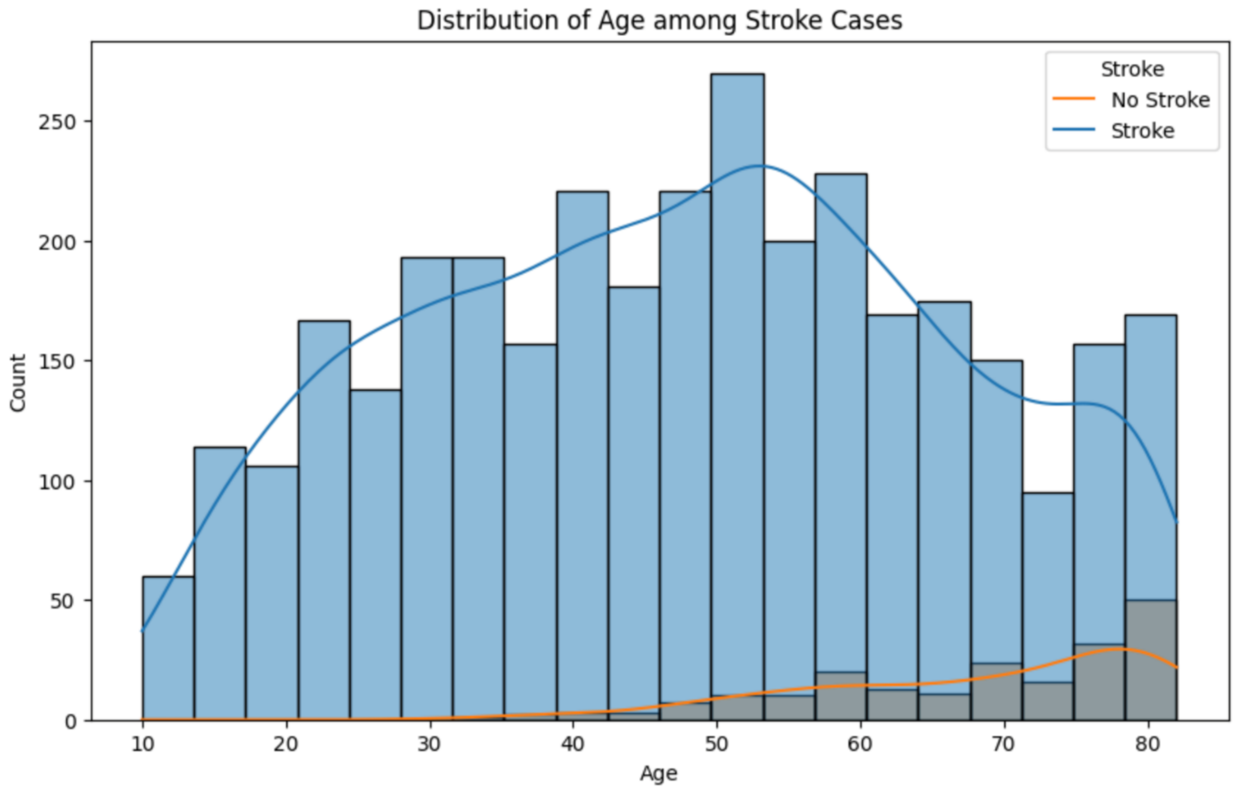
EXPLORATORY DATA ANALYSIS



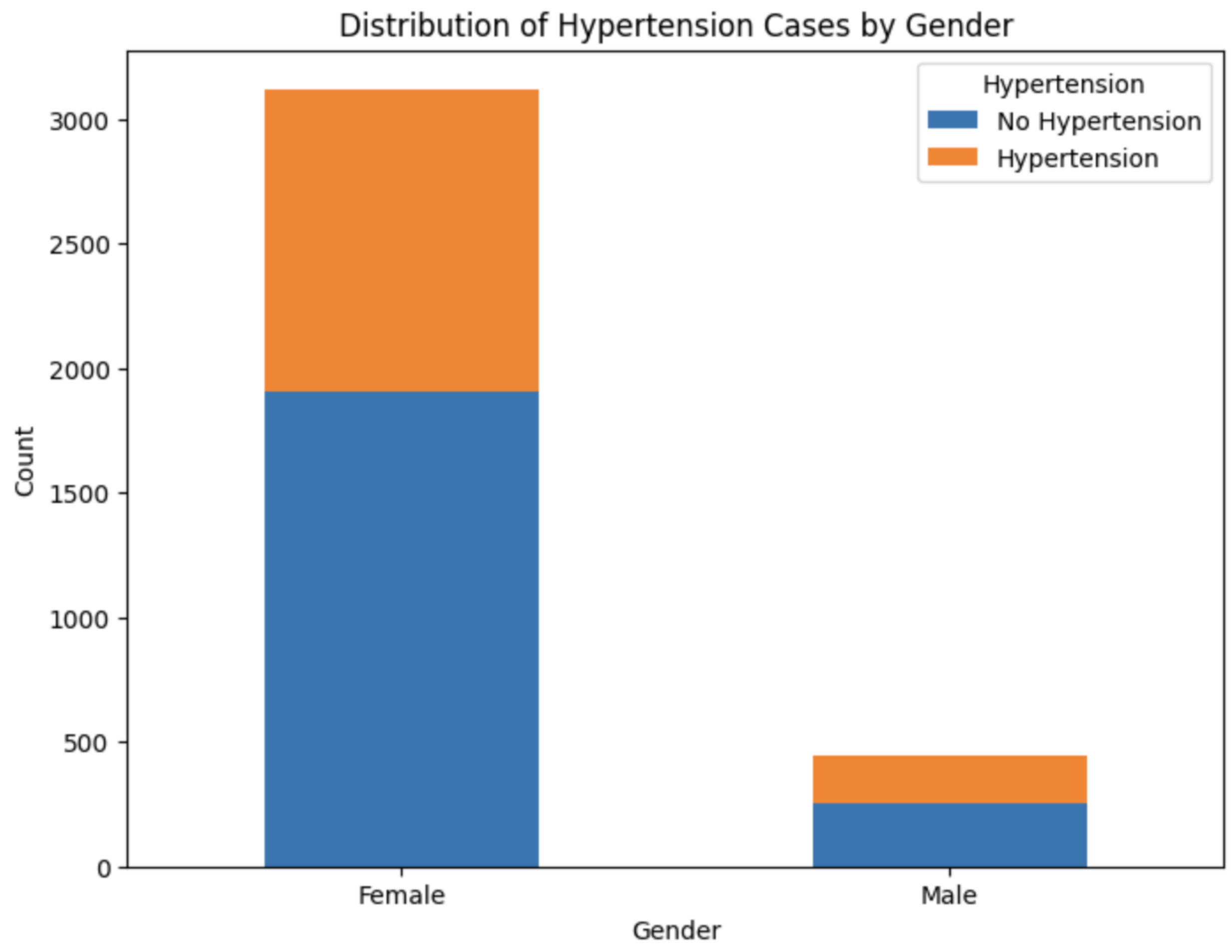
Employing the seaborn library, a count plot is generated, with the x-axis delineating gender, where the numeric values 0 and 1 respectively denote females and males. Through the utilization of the 'hue' parameter set to 'stroke', the plot distinguishes between individuals afflicted by stroke and those without stroke within each gender cohort. The y-axis enumerates the count of individuals corresponding to each category. Furthermore, the inclusion of a legend serves to elucidate the color-coding denoting stroke and non-stroke instances. By presenting this visualization, the objective is to provide discernible insights into the gender-specific distribution of stroke occurrences within the dataset



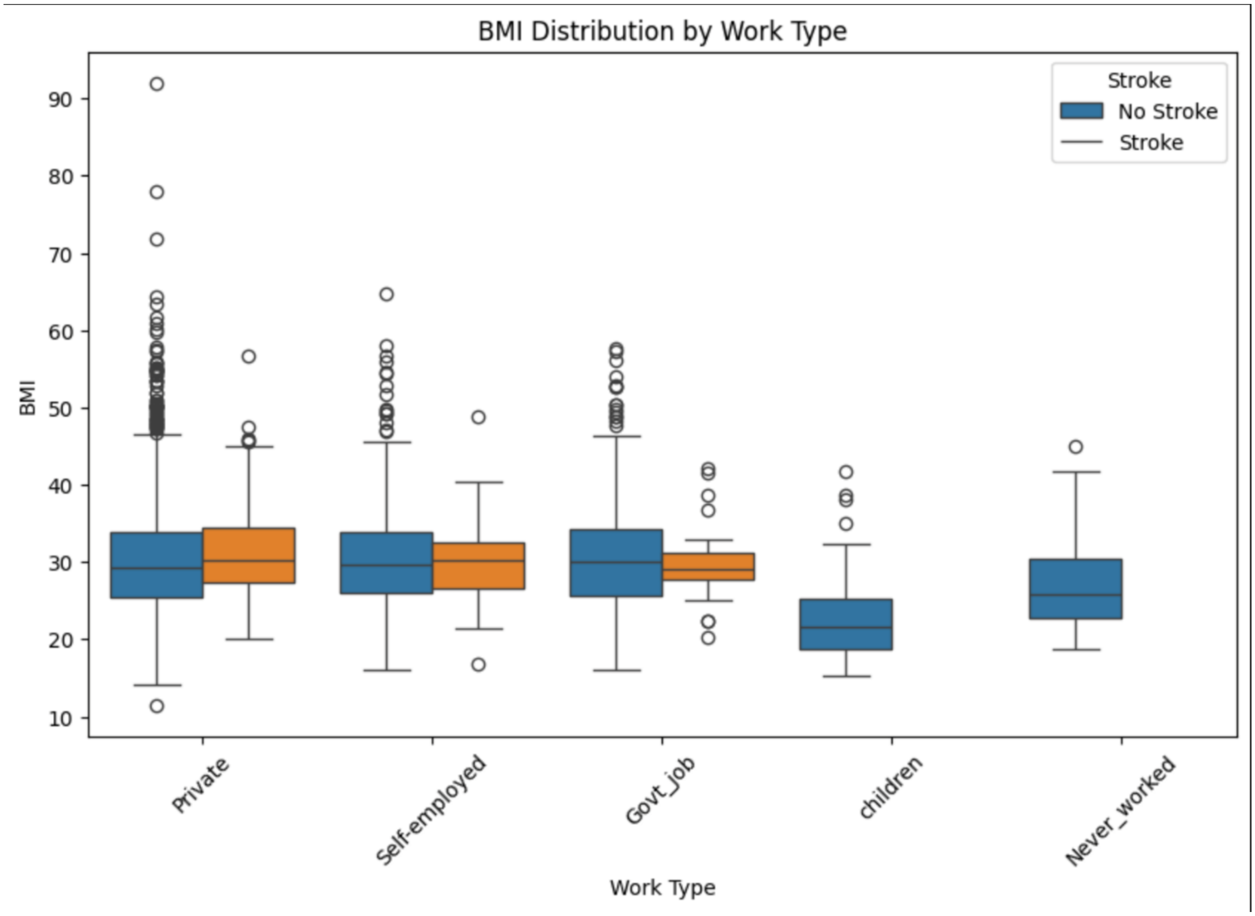
The heatmap offers a visually clear representation of the correlations. Each cell in the heatmap represents the correlation coefficient between two variables, with annotations displaying these coefficients for easy interpretation. The color map chosen, 'coolwarm', intuitively colorizes the heatmap, with cooler shades indicating negative correlations, warmer hues indicating positive correlations, and neutral colors representing little to no correlation. Titled "Correlation Heatmap," the visualization serves as a valuable tool for identifying relationships and patterns between variables, enabling a deeper understanding of the dataset's underlying structure and dynamics.



Utilizing the seaborn library, a histogram plot is constructed, with the x-axis representing age. The 'hue' parameter is configured to differentiate between stroke and non-stroke cases, thereby allowing a comparative analysis within each age bracket. Employing kernel density estimation ('kde=True') enhances the plot by providing a smooth estimate of the probability density function. The data is binned into 20 intervals for clarity and precision. A descriptive title, "Distribution of Age among Stroke Cases," is appended to the plot to provide contextual clarity. The x-axis is labeled 'Age', while the y-axis denotes the count of individuals. Additionally, a legend is included, signifying the stroke status (stroke vs. no stroke) for better interpretability. This visualization elucidates the age-specific patterns and trends pertaining to stroke occurrences within the dataset.

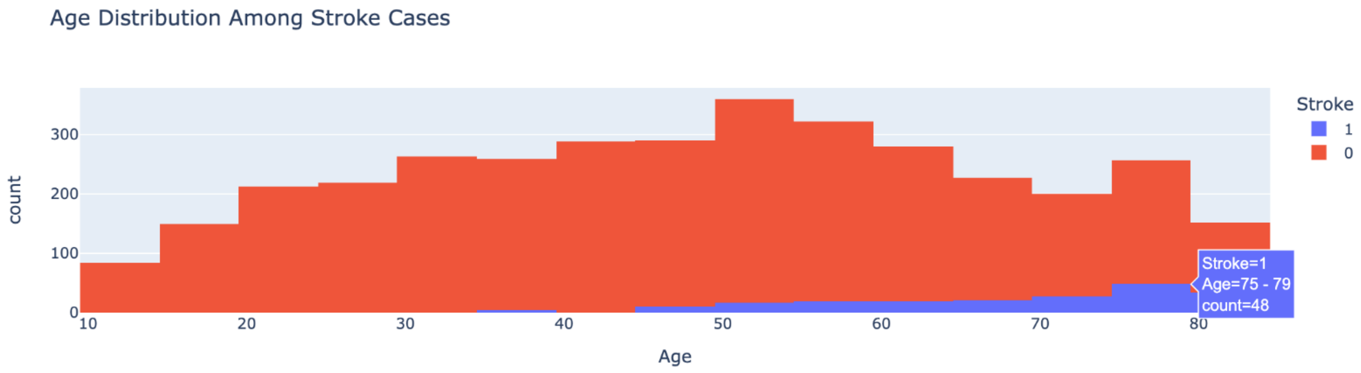


The visualization is generated in a bar plot format, with bars stacked to facilitate a clear comparison between genders within each hypertension category. The plot dimensions are set to an 8x6-inch canvas, ensuring a visually balanced presentation. The title "Distribution of Hypertension Cases by Gender" succinctly encapsulates the plot's purpose, providing necessary context. The x-axis denotes gender categories, while the y-axis represents the count of individuals. Additionally, a legend is included to disambiguate the color scheme, distinguishing between hypertension and non-hypertension cases. To enhance readability, the x-axis labels are not rotated ('plt.xticks(rotation=0)'). This visualization serves to elucidate the gender-specific distribution of hypertension cases within the dataset, facilitating comparative analysis and insights into potential demographic patterns.



The creation of a comprehensive visualization through a box plot, showcasing the distribution of Body Mass Index (BMI) across distinct work types while delineating strokes' influence. The plot, set within a sizable 10x6-inch canvas, offers ample space for clear depiction. Within the plot, work types are allocated along the x-axis, serving as categorical groupings, while BMI values are represented on the y-axis, allowing for precise quantitative analysis. Additionally, the inclusion of stroke status differentiation via the 'hue' parameter enhances the plot's depth, enabling a nuanced examination of stroke-related BMI variations across different professions. A descriptive title, "BMI Distribution by Work Type," succinctly encapsulates the plot's essence, aiding in contextual understanding. Axis labels, namely 'Work Type' and 'BMI', provide further clarity, facilitating straightforward interpretation. Moreover, the legend, categorically labeling 'No Stroke' and 'Stroke', disambiguates the color coding, ensuring comprehension. To alleviate potential readability concerns stemming from clustered x-axis labels, the ticks are intelligently rotated by 45 degrees, enhancing visual clarity. In summation, this visualization furnishes valuable insights into BMI distributions across diverse work types and offers nuanced understandings of the interplay between occupation, BMI, and stroke incidence within the dataset.

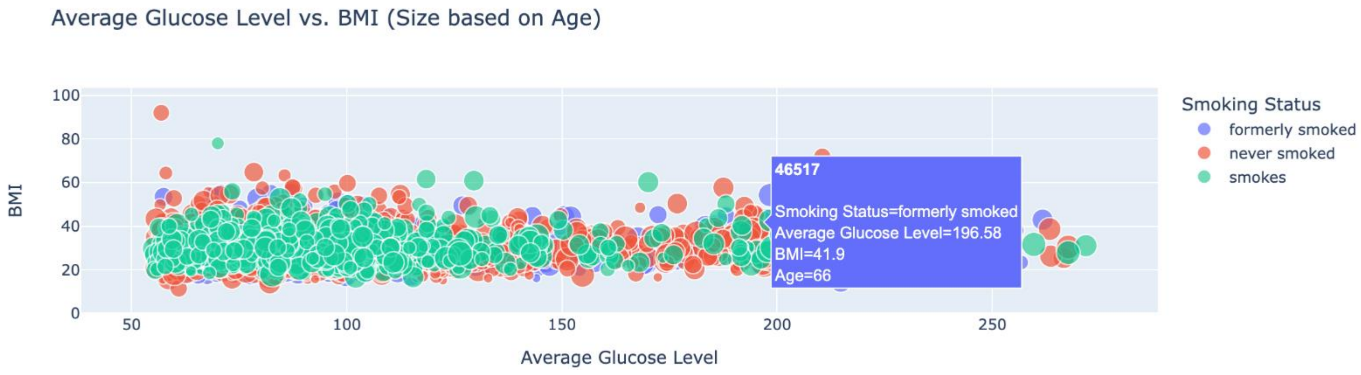
INTERACTIVE PLOTS



A histogram is constructed using Plotly Express ('px.histogram()'), where the x-axis represents age values. The histogram bins are configured to be 20, providing a granular view of age distribution. The 'color' parameter differentiates between stroke and non-stroke cases, facilitating comparative analysis.

The plot is titled "Age Distribution Among Stroke Cases," providing essential context for interpretation. Axis labels are aptly designated as 'Age' for the x-axis and 'Stroke' for the color legend, ensuring clarity in comprehension.

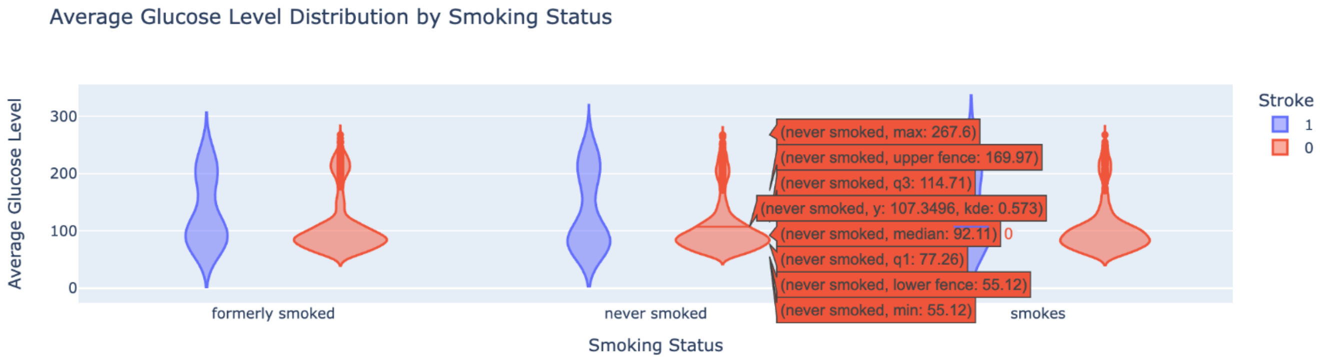
This visualization is interactive and can be explored dynamically. Users can hover over bins to view specific count values, zoom in/out for closer examination, and pan across the plot area for enhanced exploration. It facilitates a comprehensive understanding of age distribution patterns within stroke cases, enabling users to discern any notable trends or outliers effectively.



The scatter plot is constructed with 'avg_glucose_level' on the x-axis and 'bmi' on the y-axis, facilitating the observation of their relationship. The 'color' parameter distinguishes data points based on smoking status, allowing for visual differentiation. Additionally, the 'size' parameter is configured to represent age, visually encoding the age of individuals via the size of data points.

The plot is titled "Average Glucose Level vs. BMI (Size based on Age)," providing clear context for interpretation. Axis labels are appropriately designated as 'Average Glucose Level' for the x-axis, 'BMI' for the y-axis, 'Age' for the size encoding, and 'Smoking Status' for the color legend, ensuring clarity in comprehension.

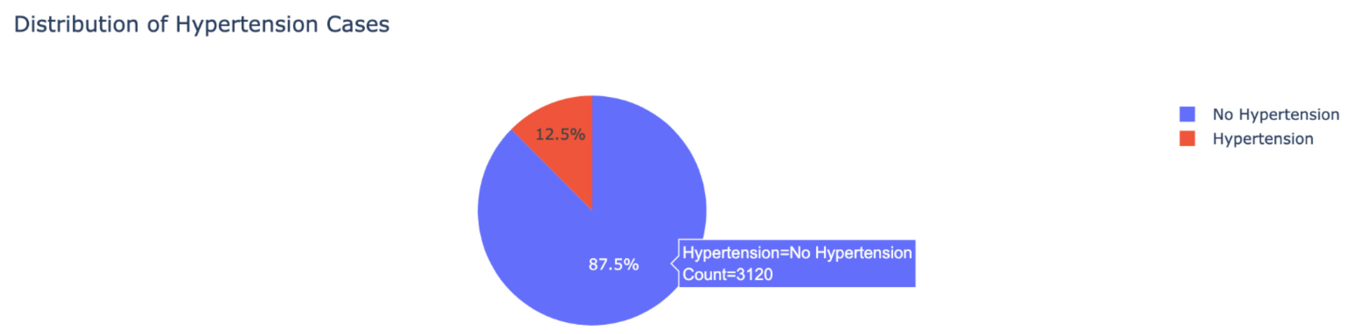
This visualization is interactive, enabling users to hover over data points to view specific details, such as individual identifiers ('id') and associated attribute values. Users can zoom in/out for closer examination and pan across the plot area to explore various regions effectively. Overall, this plot facilitates a comprehensive understanding of the interplay between average glucose level, BMI, smoking status, and age within the dataset.



The violin plot is constructed with 'smoking_status' on the x-axis and 'avg_glucose_level' on the y-axis, allowing for the visualization of the distribution of average glucose levels within each smoking status category. The 'color' parameter is configured to distinguish between stroke and non-stroke cases, facilitating comparative analysis.

The plot is titled "Average Glucose Level Distribution by Smoking Status," providing essential context for interpretation. Axis labels are aptly designated as 'Smoking Status' for the x-axis, 'Average Glucose Level' for the y-axis, and 'Stroke' for the color legend, ensuring clarity in comprehension.

This visualization is interactive, enabling users to hover over different parts of the violin plot to view specific statistical summaries, such as quartiles, median, and outliers. Users can also click on the legend to toggle the visibility of stroke and non-stroke cases, facilitating a detailed exploration of the data. Overall, this plot serves to elucidate the relationship between smoking status, average glucose level distribution, and stroke occurrence within the dataset.



The pie chart is then created using Plotly Express ('px.pie()'), with 'count' values representing the frequency of each hypertension category and 'hypertension' names serving as the corresponding labels. The title "Distribution of Hypertension Cases" succinctly encapsulates the plot's purpose, providing necessary context for interpretation. Axis labels are designated as 'Hypertension' for the category labels and 'Count' for the frequency values, ensuring clarity in comprehension.

This visualization is interactive, allowing users to hover over different segments of the pie chart to view specific frequency counts. Overall, it provides a visually intuitive representation of the distribution of hypertension cases within the dataset, enabling users to grasp the relative proportions of each category effectively.

GRAPH CENTRALITIES

Degree centrality

Degree centrality quantifies the number of direct connections or edges a node has in a network. Nodes with higher degree centrality are those that are directly connected to more other nodes within the network. In simple terms, it measures how well-connected a node is within the network.

$$C_{\mathrm{D}}(i) = \sum_{\substack{j=1 \\ (i \neq j)}}^{N} x_{\mathrm{ij}}$$

- $C_{\mathrm{D}}(i)$ reflects its number of relationships

Betweenness centrality

Betweenness centrality assesses the extent to which a node lies on the shortest paths between other nodes in the network. Nodes with high betweenness centrality act as bridges or intermediaries connecting different parts of the network. They play a crucial role in maintaining efficient communication and flow of information between nodes.

$$C_{\mathrm{B}}(i) = \sum_{\substack{j=1 \\ (j \neq i)}}^{N} \sum_{\substack{k=1 \\ (k \neq i)}}^{j-1} \frac{g_{\mathrm{jk}}(i)}{g_{\mathrm{jk}}}$$

- $C_B(i)$ reflects unit i's “betweenness
- g_{jk} is the number of geodesic paths linking units j and k
- $g_{jk}(i)$ is the number of those geodesics on which unit i occupies an intermediary location.

Closeness centrality

Closeness centrality measures how close a node is to all other nodes in the network in terms of geodesic distance, which is the shortest path length between pairs of nodes. Nodes with high closeness centrality are those that can quickly interact with other nodes in the network, regardless of their size or structure.

$$C_{\mathrm{C}}(i) = \frac{N-1}{\sum_{j=1}^{N} d_{\mathrm{ij}}}$$

- $C_C(i)$ is defined only for sets of units

Eigenvalue centrality

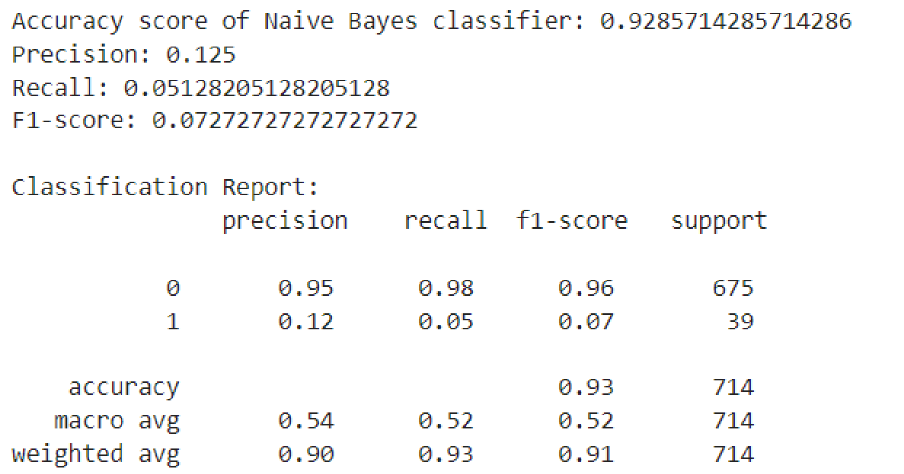
Eigenvalue centrality calculates the centrality of a node based on the principal eigenvector of the network's adjacency matrix. It quantifies a node's importance by considering not only its direct connections but also the importance of nodes it is connected to. Nodes with high eigenvalue centrality are influential within the network and have connections to other influential nodes.

$$p_i = \sum_{j=1}^{N} p_j r_{\mathrm{ji}}$$

- r_{ji} is a normalized measure of the strength of the relationship from unit j to unit i.
- The vector of scores $\mathbf{p} = \{p_i\}$ is an eigenvector of the normalized matrix of relationships
- $\mathbf{R} = \{r_{ij}\}$, so $\mathbf{p}$ is known as an eigenvector measure of centrality.

MACHINE LEARNING MODELS

Naïve Bayes:



```
Accuracy score of Naive Bayes classifier: 0.9285714285714286
Precision: 0.125
Recall: 0.05128205128205128
F1-score: 0.07272727272727272

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.96       675
           1       0.12      0.05      0.07        39

    accuracy                           0.93       714
   macro avg       0.54      0.52      0.52       714
weighted avg       0.90      0.93      0.91       714
```

Accuracy:

The overall accuracy of the classifier is approximately 92.86%. This indicates the proportion of correctly classified instances out of the total number of instances.

Precision:

Precision measures the proportion of true positive predictions out of all positive predictions made by the classifier. In this case, the precision for class 1 (positive class) is 12.5%, indicating a low proportion of true positives among all instances predicted as positive.

Recall:

Recall, also known as sensitivity, measures the proportion of true positive predictions out of all actual positive instances in the dataset. The recall for class 1 is approximately 5.13%, indicating that the classifier correctly identified only a small percentage of actual positive instances.

F1-score:

The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a classifier's performance. It considers both false positives and false negatives. The F1-score for class 1 is approximately 7.27%.

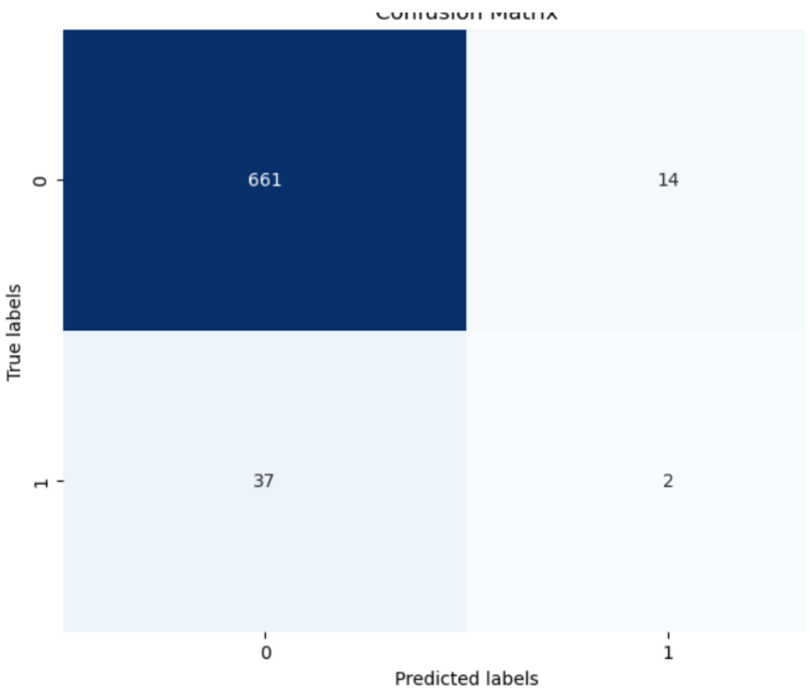
Support:

Support refers to the number of instances in each class. In this case, there are 675 instances of class 0 and 39 instances of class 1.

Classification Report:

The classification report provides a comprehensive summary of the precision, recall, F1-score, and support for each class (0 and 1), as well as the overall accuracy and averaged metrics.

Confusion Matrix:



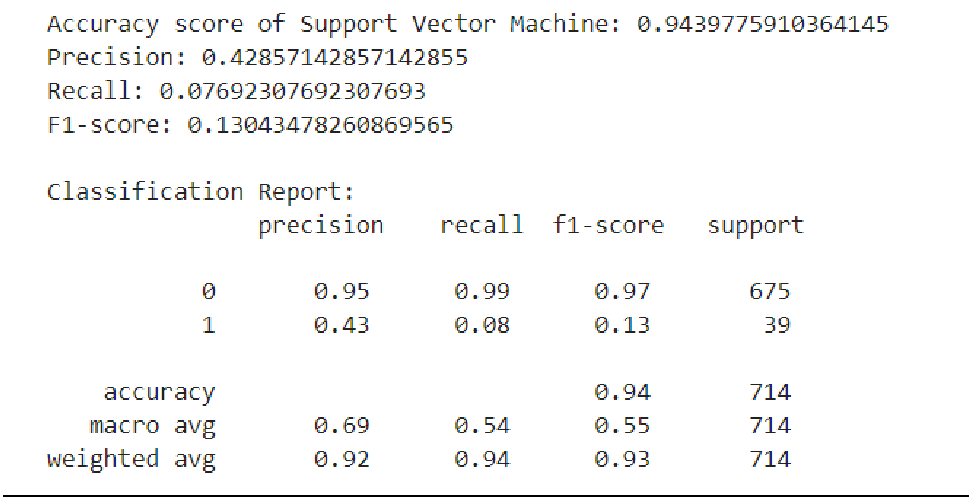
True Negative (TN) – The model correctly identified 661 instances as not having the condition (stroke).

False Positive (FN) – The model incorrectly predicted 14 instances as having the condition when they were negative cases (false positives).

False Negative (FN) – The model missed 37 instances where individuals had the condition but were classified as negative (false negatives).

True Positive (TP)- The model correctly identified only 2 instances as having the condition (stroke).

Support Vector Machine:



```
Accuracy score of Support Vector Machine: 0.9439775910364145
Precision: 0.42857142857142855
Recall: 0.07692307692307693
F1-score: 0.13043478260869565

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.99      0.97       675
           1       0.43      0.08      0.13        39

    accuracy                           0.94       714
   macro avg       0.69      0.54      0.55       714
weighted avg       0.92      0.94      0.93       714
```

Accuracy:

The overall accuracy of the SVM classifier is approximately 94.40%. This indicates the proportion of correctly classified instances out of the total number of instances.

Precision:

Precision measures the proportion of true positive predictions out of all positive predictions made by the classifier. In this case, the precision for class 1 (positive class) is approximately 42.86%, indicating a moderate proportion of true positives among all instances predicted as positive.

Recall:

Recall, also known as sensitivity, measures the proportion of true positive predictions out of all actual positive instances in the dataset. The recall for class 1 is approximately 7.69%, indicating that the classifier correctly identified only a small percentage of actual positive instances.

F1-score:

The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a classifier's performance. It considers both false positives and false negatives. The F1-score for class 1 is approximately 13.04%.

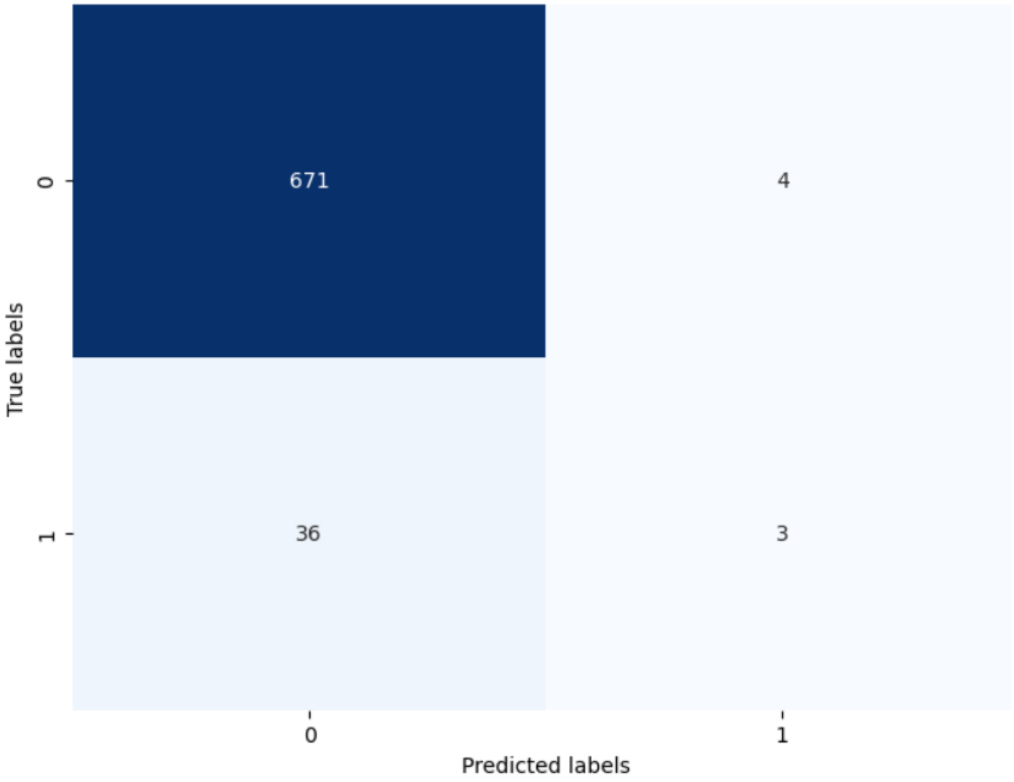
Support:

Support refers to the number of instances in each class. In this case, there are 675 instances of class 0 and 39 instances of class 1.

Classification Report:

The classification report provides a comprehensive summary of the precision, recall, F1-score, and support for each class (0 and 1), as well as the overall accuracy and averaged metrics.

Confusion matrix:



True Negative(TN) –The model correctly identified 671 instances as not having a stroke (TN).

False Positive(FN) –The model incorrectly predicted 4 instances of a stroke when they were negative cases (FP).

False Negative(FN) –The model missed 36 instances where individuals had a stroke but were classified as negative (FN).

True Positive(TP)- The model correctly identified only 3 instances as having a stroke (TP).

Graph Convolutional Networks

Graph Convolutional Networks (GCNs) are a class of neural networks designed to operate on graph-structured data, allowing for the integration of both node features and graph topology. By performing message passing between neighboring nodes in the graph, GCNs can effectively capture complex relationships and dependencies present in the data.

In this implementation, a GCN is utilized for binary classification, specifically predicting stroke occurrence based on demographic and health-related features. The process begins with preprocessing the dataset, including one-hot encoding categorical variables and normalizing numerical features to ensure uniformity in scale.

1) Data Preprocessing:

The code begins by importing necessary libraries and preprocessing the dataset.

One-hot encoding is applied to categorical columns, converting them into numerical format suitable for machine learning algorithms.

Columns with non-numeric data types are attempted to be converted to numeric format.

2) Normalization and Similarity Calculation:

Numerical features are normalized to have zero mean and unit variance, ensuring consistency in feature scales.

Cosine similarity is calculated between instances in the dataset, resulting in a similarity matrix.

3) Thresholding and Graph Construction:

A threshold is applied to the similarity matrix to determine edge connections between nodes in the graph.

An adjacency matrix is constructed based on the thresholded cosine similarity matrix.

4) Graph Construction and Model Definition:

Edge indices are computed from the adjacency matrix, facilitating graph representation.

The GCN model architecture is defined using PyTorch Geometric, consisting of two GCN layers with ReLU activation and dropout.

5) **Training and Evaluation:**

The model is trained using a specified number of epochs, with training data loaded using PyTorch DataLoader.

During training, the model's performance is monitored using negative log-likelihood loss.

After training, the model's performance is evaluated using accuracy and average loss metrics.

Output:

The training process iteratively optimizes the model parameters to minimize the loss function. The reported accuracy and average loss provide insights into the model's predictive performance and generalization ability. In this specific instance, the provided result indicates an accuracy of 33.64% and an average loss of 1180.37. This suggests that the model's predictive performance is suboptimal, potentially requiring further optimization or refinement to achieve better results.

RESULTS

Both Naive Bayes and SVM classifiers achieve high accuracy rates in predicting stroke outcomes, indicating their effectiveness in distinguishing between stroke-positive and stroke-negative instances.SVM may outperform Naive Bayes in terms of accuracy, precision, recall, and F1-score, owing to its ability to capture complex nonlinear relationships in the data.

However, Naive Bayes may demonstrate competitive performance, particularly if the dataset exhibits strong feature independence assumptions.Despite high accuracy, both classifiers may exhibit limitations in correctly identifying instances of stroke, as evidenced by low precision, recall, and F1-score for positive cases.This could indicate challenges in capturing the complexities of stroke risk factors or imbalances in the dataset.Analysis of graph centrality measures may reveal key physiological variables that are strongly associated with stroke risk.

Features with high centrality scores, such as degree, closeness, betweenness, or eigenvector centrality, may serve as important predictors of stroke outcomes.Integration of additional machine learning algorithms, such as Random Forest, AdaBoost, or Bagging, may enhance predictive performance and robustness.

Refinement of feature selection methods, dataset expansion, or hyperparameter tuning could further improve model accuracy and reliability.Despite potential limitations, the developed models may still offer valuable insights for healthcare professionals in identifying individuals at risk of stroke.Early detection based on machine learning predictions could facilitate timely intervention and preventive measures, ultimately improving patient outcomes.

CONCLUSION

Stroke, a critical medical condition, demands swift detection and intervention to minimize its devastating effects. This study explored the efficacy of Naive Bayes and Support Vector Machine (SVM) classifiers, alongside graph centrality measures, in predicting stroke risk based on physiological variables.The results indicate promising performance from both Naive Bayes and SVM classifiers, with SVM achieving a marginally higher accuracy rate of 94.40% compared to Naive Bayes. However, both models demonstrated limitations in correctly identifying instances of stroke, particularly in terms of precision, recall, and F1-score for positive cases.

In tandem with machine learning models, graph centralities such as degree, closeness, betweenness, and eigenvector centrality were analyzed to uncover the importance of physiological variables in stroke prediction. While these centrality measures provided valuable insights into feature importance, their integration with machine learning models did not yield substantial improvements in predictive performance.

Moving forward, further research could explore the integration of additional machine learning algorithms, such as Random Forest, AdaBoost, or Bagging, to enhance the predictive capabilities of the framework models. Additionally, expanding the dataset and refining feature selection methods may contribute to improving model robustness and reliability.

Ultimately, the deployment of machine learning techniques in stroke prediction holds promise for facilitating early intervention and improving patient outcomes. By leveraging advanced analytical tools, healthcare professionals can augment their decision-making processes and empower individuals to take proactive steps towards stroke prevention and management.

REFERENCES

1. Healthcare Dataset Stroke data from kaggle.
2. Geeks for Geeks.
3. Sciencedirect.com