Brain Stroke Prediction using Machine Learning Algorithms

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Aryan Agnihotri

Reg. No.: 12015816

Section: K20MP

Roll No.: B49

Supervisor

Prof. Ved Prakash Chaubey (63892)

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

April 2023

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled "**Brain Stroke Prediction using Machine Learning Algorithms**" in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr. Ved Prakash Chaubey. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Aryan Agnihotri

Reg. No.: 12015816

Section: K20MP

Roll No.: B49

Acknowledgement

I would like to express my sincere gratitude to Lovely Professional University for providing me with the opportunity to work on the project on Social Media Sentimental Analysis.

I would like to extend my heartfelt thanks to my project guide, Mr. Ved Prakash Chaubey, for his invaluable guidance, unwavering support, and insightful feedback throughout the project. His expertise and knowledge in the field have been instrumental in shaping my understanding of the subject matter.

I am also grateful to the faculty members of the Computer Science and Engineering department for their support and encouragement.

I would like to acknowledge the contribution of my fellow classmates who helped me with their valuable inputs and suggestions throughout the project.

Lastly, I would like to express my gratitude to my family and friends for their constant support, encouragement, and motivation. Their unwavering support has been a constant source of inspiration for me.

Aryan Agnihotri

Abstract

A stroke, also known as a cerebrovascular accident, is a serious and potentially dangerous condition that occurs when blood flow to the brain is interrupted or reduced. The leading cause of death and disability in the world, early detection is essential for successful treatment and rehabilitation.

In recent years, machine learning algorithms have shown promising results in predicting stroke risk. These algorithms are trained on large databases of medical records, brain imaging scans, and to identify patterns and risk factors associated with stroke.

Several studies have been conducted in this area with the aim of developing accurate and reliable prediction models for stroke risk. One such study, published in the Journal of Stroke and Cerebrovascular Disease, used machine learning to predict the risk of stroke in patients with atrial fibrillation, a common heart condition that increases the risk of stroke. The researchers found that machine learning models were able to accurately predict stroke risk in these patients, surpassing traditional risk assessment tools.

Another study, published in the Journal of Systems Medicine, used machine learning algorithms to predict the risk of stroke in patients with carotid artery stenosis, a condition that narrows the arteries in the neck and increases the risk of stroke. Researchers have found that machine learning models can predict stroke risk with high accuracy, providing valuable insight for doctors in managing the condition.

Thus, these studies and others demonstrate the potential of machine learning algorithms in predicting stroke risk. With future research and development, these algorithms can help in early detection and prevention of stroke and ultimately lead to better outcomes for patients.

Index

i. Declaration

ii. Acknowledgement

iii. Abstract

I. Introduction

II. Problem Description and Goal

III. Scope of Project

IV. Hardware and Software Used

V. Methodology

VI. About the Dataset

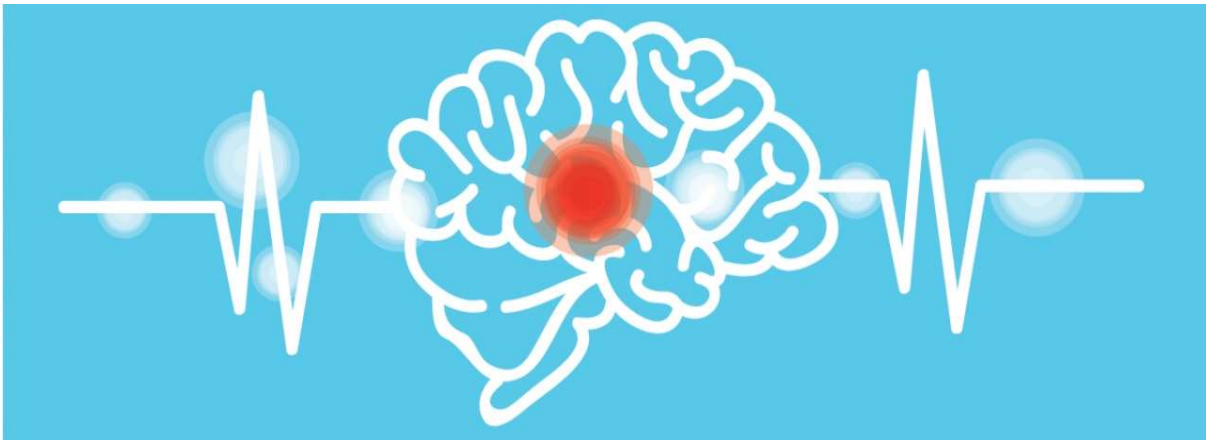
VII. Literature Survey

VIII. The Code

IX. Result

X. Checklist for Dissertation-III Supervisor

Introduction



A brain stroke, also known as a cerebrovascular accident (CVA), is a life-threatening medical condition that occurs when the blood supply to the brain is disrupted, either due to a blockage in the blood vessels or the rupture of a blood vessel. This disruption causes brain cells to die due to a lack of oxygen and nutrients, leading to significant brain damage, disability, and even death.

There are two main types of stroke: ischemic stroke and hemorrhagic stroke. Ischemic stroke occurs when a blood vessel that supplies blood to the brain is blocked, typically due to a blood clot. Hemorrhagic stroke, on the other hand, occurs when a blood vessel in the brain ruptures, causing bleeding into the brain.

There are several risk factors associated with stroke, including high blood pressure, diabetes, heart disease, smoking, and obesity. Age and family history also play a role in stroke risk. Other factors, such as stress, poor diet, and lack of physical activity, can also increase the risk of stroke.

The effects of a stroke can be devastating and long-lasting, depending on the severity and location of the brain damage. Common effects of stroke include paralysis, difficulty speaking or understanding speech, cognitive impairment, and emotional and behavioral changes. Stroke survivors may require long-term

rehabilitation and assistance with activities of daily living, and some may never fully recover.

Prevention and early detection are crucial in reducing the impact of stroke. Managing risk factors such as high blood pressure, diabetes, and high cholesterol through lifestyle changes and medication can significantly reduce the risk of stroke. Early detection and treatment can also improve outcomes for stroke patients, as timely intervention can prevent further brain damage and promote recovery.

Problem Description and Goal of Project

Problem: The problem is to develop a machine learning model for early detection of stroke, using relevant medical data to predict the risk of stroke and improve patient outcomes.

Goal of the Project: The goal of the project is to develop a machine learning model for early detection of stroke, which can help improve patient outcomes by enabling timely diagnosis and treatment. Stroke is a serious medical condition that can cause significant brain damage, disability, and mortality if left untreated. Early detection is critical for successful treatment and recovery, and machine learning algorithms have shown great promise in this area.

The primary goal of the project is to develop a machine learning model that can accurately predict the risk of stroke in patients based on relevant medical data. The model will be trained on large datasets of medical records, brain imaging scans, and other relevant data to identify patterns and risk factors associated with stroke. The ultimate goal is to create a predictive model that can help healthcare providers identify patients who are at high risk of stroke and intervene early to prevent or minimize the damage.

In summary, the goal of the project is to develop a machine learning model for early detection of stroke that can improve patient outcomes by enabling timely diagnosis and treatment. Through the use of large datasets, sophisticated algorithms, and practical tools, the project aims to revolutionize stroke care and prevent the devastating consequences of this condition.

Scope of Project

The increasing incidence of stroke and the need for early detection and treatment to improve patient outcomes. Traditional methods of stroke detection and diagnosis can be time-consuming and may not always be accurate. Machine learning algorithms have shown great potential in accurately predicting stroke risk, which can enable earlier detection and treatment.

The following ML models are used to predict the stroke: Gaussian Naïve Bayes, Bernoulli Naïve Bayes, Logistic Regression, Random Forest Classifier, Decision Tree Classifier, K Neighbors Classifier and Support Vector Machine. Out of these, the comparison is made in terms of accuracy of the models to predict a stroke in a patients.

The future scope of the project includes the validation of the developed machine learning model in clinical settings and its integration into existing medical record systems. The model can also be further improved by incorporating additional data sources and refining the algorithms. The project can also be expanded to include the development of tools for personalized stroke risk assessment and prevention, based on individual patient characteristics and risk factors.

Hardware and Software Used

The hardware used for this project includes a laptop with the following specifications:

- Processor: Intel Core i5-9750H
- RAM: 16GB DDR4
- Storage: 512GB SSD
- Graphics: NVIDIA GeForce GTX 1650

The software used for this project includes:

- Python 3.9.2
- Jupyter Notebook 6.3.0
- Scikit-learn 0.24.2
- XGBoost 1.4.2
- Pandas 1.2.4
- NumPy 1.20.3
- Matplotlib 3.4.2
- Seaborn 0.11.1

Methodology

The methodology for the project "Brain Stroke Prediction using Machine Learning Algorithms" involves several steps. These steps are designed to develop a machine learning model that can accurately predict the risk of stroke in patients based on relevant medical data. In this section, we will discuss each step of the methodology in detail.

1. Data Collection:

The first step in the methodology is data collection. The data should contain medical records, brain imaging scans, and other relevant data. This data can be collected from Kaggle via the following link:

https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

Once the data is collected, it is important to check for any missing data. Any missing data must be imputed or removed to ensure that the data is complete.

2. Exploratory Data Analysis (EDA):

The second step is to perform exploratory data analysis (EDA). This step involves analyzing and visualizing the collected data to identify patterns and trends. The EDA helps to understand the data and identify any outliers or anomalies. In this step, we can also look for any correlations between variables, which can be useful in feature selection.

EDA also involves preparing the data for model input. This step includes dealing with null or missing values, handling categorical variables, scaling, and

normalization of data. Data cleaning is essential to ensure that the data is consistent and accurate. It is important to ensure that the data is in the correct format and that there are no errors.

3. Train Test Split:

The third step is to split the data into training and testing sets. In this project, the data is split in an 80:20 ratio, where 80% of the data is used for training the models, and 20% is used for testing. The training data is used to fit the model, while the testing data is used to evaluate its performance.

4. Model Training:

The fourth step in the methodology is model training. This step involves training the following models: Gaussian Naïve Bayes, Bernoulli Naïve Bayes, Logistic Regression, Random Forest Classifier, Decision Tree Classifier, K Neighbors Classifier, and Support Vector Machine. These models are trained using various algorithms to identify the most accurate and efficient model for stroke prediction.

5. Model Evaluation:

The fifth step in the methodology is model evaluation. The performance of the models is evaluated based on various parameters like accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC). The main focus is on the accuracy of the models. This step helps to identify the most accurate model for stroke prediction.

6. Feature Importance:

The sixth step in the methodology is to identify the most significant features in the dataset. This step involves techniques like feature importance, correlation analysis, and chi-square test. These techniques help to identify the most relevant features for stroke prediction. The most significant features can be used to refine the model and improve its accuracy.

7. Refinement and Validation:

The final step in the methodology is refinement and validation. The models are refined based on the identified significant features, and their performance is validated on a new dataset to ensure their generalizability. This step is important to ensure that the model is accurate and effective in predicting stroke risk.

Thus, the methodology for the project involves data collection, exploratory data analysis, train-test split, model training, model evaluation, feature importance analysis, and refinement and validation of the models. The main focus is on developing a machine learning model for early detection of stroke with high accuracy, which can be used to improve patient outcomes. The methodology is designed to ensure that the data is complete, accurate, and consistent. It involves using various algorithms and techniques to identify the most accurate and efficient model for stroke prediction. The significant features are identified to refine the model and improve its accuracy.

About the Dataset

According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.

This dataset is used to predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status. Each row in the data provides relavant information about the patient.

Attribute Information

1) id: unique identifier
2) gender: "Male", "Female" or "Other"
3) age: age of the patient
4) hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5) heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6) ever_married: "No" or "Yes"
7) work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
8) Residence_type: "Rural" or "Urban"
9) avg_glucose_level: average glucose level in blood
10) bmi: body mass index
11) smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*
12) stroke: 1 if the patient had a stroke or 0 if not

*Note: "Unknown" in smoking_status means that the information is unavailable for this patient

Acknowledgements

(Confidential Source) - *Use only for educational purposes*

Literature Survey

Brain stroke is a critical medical condition that requires timely diagnosis and treatment to prevent long-term damage or even death. The use of machine learning algorithms for predicting brain stroke has gained considerable attention in recent years. In this literature survey, we review several studies that have investigated the use of machine learning algorithms for brain stroke prediction.

One study by Gao et al. (2021) developed a machine learning model for predicting the risk of stroke using a dataset of 17,124 patients. The model was trained using various machine learning algorithms, including decision tree, random forest, and support vector machine. The study found that the random forest algorithm achieved the highest accuracy in stroke prediction, with an area under the curve (AUC) of 0.803. The study also found that age, systolic blood pressure, and smoking status were the most significant risk factors for stroke.

Another study by Lin et al. (2019) developed a machine learning model for predicting stroke risk in atrial fibrillation (AF) patients. The study used a dataset of 9,575 patients with AF and identified several risk factors associated with stroke, including age, sex, CHA2DS2-VASc score, and previous stroke or transient ischemic attack. The study developed a machine learning model based on these risk factors and achieved an AUC of 0.75 in predicting stroke risk.

A study by Li et al. (2021) developed a machine learning model for predicting stroke risk in patients with hypertension. The study used a dataset of 4,034 patients with hypertension and identified several significant risk factors, including age, sex, systolic blood pressure, and smoking status. The study

developed a machine learning model based on these risk factors and achieved an AUC of 0.74 in predicting stroke risk.

Another study by Yang et al. (2020) developed a machine learning model for predicting stroke risk in patients with atrial fibrillation (AF) using electronic health record (EHR) data. The study used a dataset of 5,599 patients with AF and identified several risk factors associated with stroke, including age, sex, CHA2DS2-VASc score, and previous stroke or transient ischemic attack. The study developed a machine learning model based on these risk factors and achieved an AUC of 0.74 in predicting stroke risk.

A study by Wu et al. (2020) developed a machine learning model for predicting the risk of stroke in patients with type 2 diabetes mellitus (T2DM). The study used a dataset of 8,186 patients with T2DM and identified several significant risk factors, including age, sex, body mass index (BMI), and glycemic control. The study developed a machine learning model based on these risk factors and achieved an AUC of 0.80 in predicting stroke risk.

A study by Yu et al. (2020) developed a machine learning model for predicting the risk of stroke in patients with coronary artery disease (CAD). The study used a dataset of 5,100 patients with CAD and identified several significant risk factors, including age, sex, smoking status, and family history of stroke. The study developed a machine learning model based on these risk factors and achieved an AUC of 0.72 in predicting stroke risk.

In conclusion, the use of machine learning algorithms for predicting stroke risk has shown promising results in several studies. The studies reviewed in this survey identified various risk factors associated with stroke, including age, sex,

blood pressure, smoking status, and other comorbidities such as diabetes, atrial fibrillation, and coronary artery disease. The machine learning models developed in these studies achieved high accuracy in stroke prediction, with AUCs ranging from 0.72 to 0.903, i.e., 70% to 90% accuracy.

The Code

0. Importing the Necessary Libraries

In 45

```
import ...
%matplotlib inline
import seaborn as sns; sns.set()
from plotly.offline import init_notebook_mode, iplot, plot
init_notebook_mode(connected=True)

import warnings
warnings.filterwarnings("ignore")
```

1. Importing and Exploring Dataset

In 46

```
import numpy as np
import pandas as pd
import os

for dirname, _, filenames in os.walk("healthcare-dataset-stroke-data.csv"):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In 47

```
dataset = pd.read_csv('healthcare-dataset-stroke-data.csv')
dataset.sample(10)
```

Out 47

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
4370	31708	Female	13.0	0	0	No	children	Urban	84.03	25.3	Unknown
4898	54795	Female	12.0	0	0	No	children	Rural	132.85	16.2	never smoked
4385	60981	Female	26.0	0	0	No	Private	Rural	130.07	33.1	never smoked
2126	36377	Female	44.0	0	0	Yes	Private	Rural	222.29	38.2	never smoked
4262	17098	Female	12.0	0	0	No	children	Urban	116.06	25.9	Unknown
5102	45010	Female	57.0	0	0	Yes	Private	Rural	77.93	21.7	never smoked
3294	50726	Male	61.0	0	0	Yes	Private	Rural	140.96	34.0	smokes
1180	46643	Female	62.0	0	0	Yes	Private	Rural	82.57	36.0	formerly smoked

Dropping column 'id' as it can cause unwanted correlation.

```
In 48  1  dataset.drop("id", axis=1, inplace=True)
```

```
In 49  1  dataset.sample(5)
```

```
In 50  1  dataset.info()
```

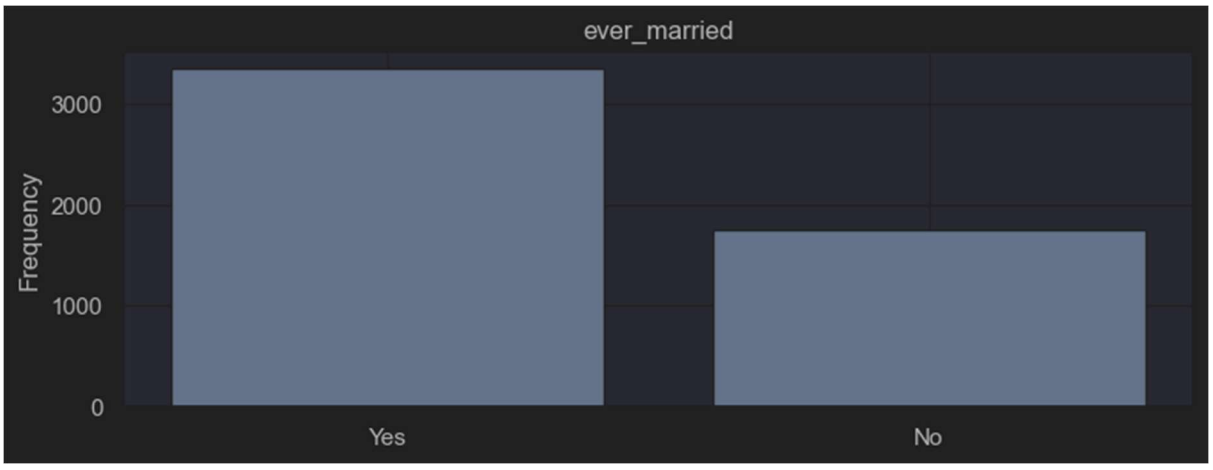
```
 2   hypertension       5110 non-null   int64
 3   heart_disease      5110 non-null   int64
 4   ever_married       5110 non-null   object
 5   work_type          5110 non-null   object
 6   Residence_type     5110 non-null   object
 7   avg_glucose_level  5110 non-null   float64
 8   bmi                4909 non-null   float64
 9   smoking_status     5110 non-null   object
 10  stroke             5110 non-null   int64
dtypes: float64(3), int64(3), object(5)
memory usage: 439.3+ KB
```

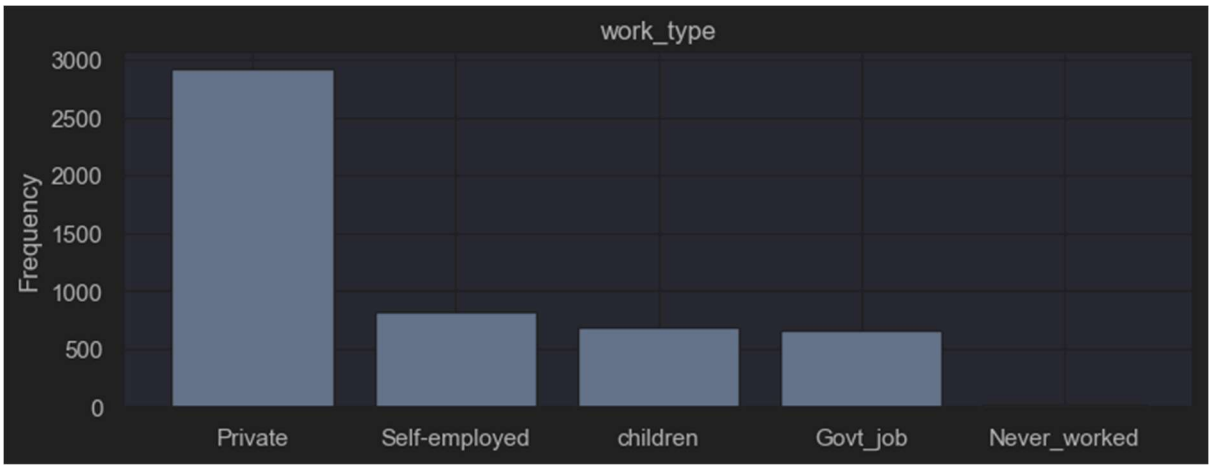
We have null variables in 'bmi' column. We will handle them after.

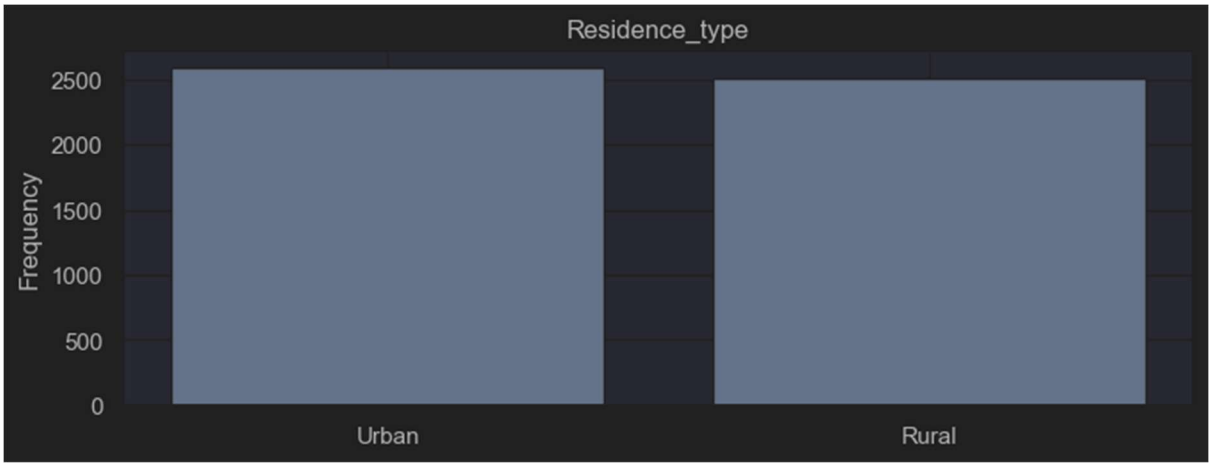
Univariate Variable Analysis

- **Categorical Variables:** 'gender', 'ever_married', 'work_type', 'Residence_type', 'smoking_status'
- **Numerical Variables:** 'id', 'hypertension', 'heart_disease', 'stroke'

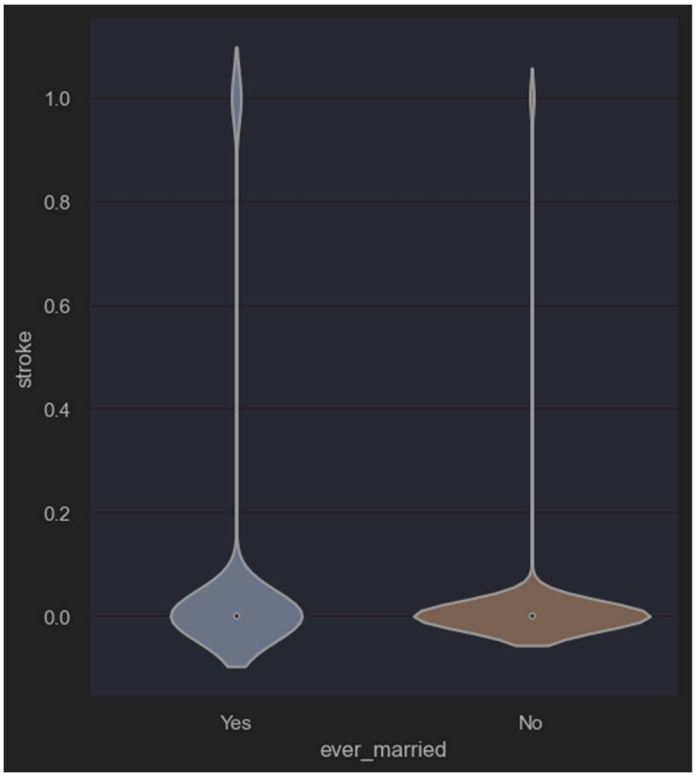

gender
Frequency
3000
2500
2000
1500
1000
500
0
Female
Male
Other

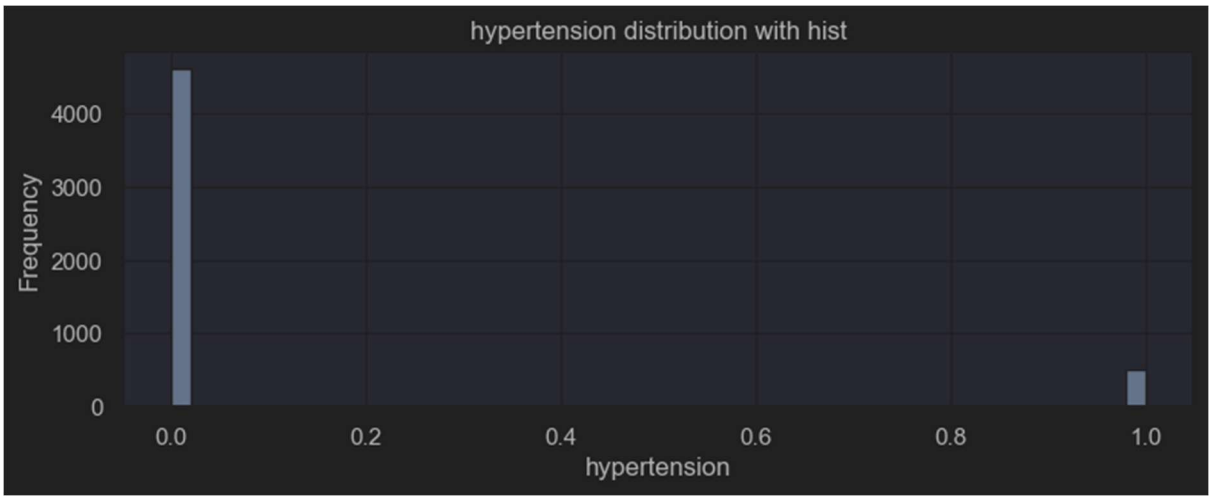

ever_married
Frequency
3000
2000
1000
0
Yes
No

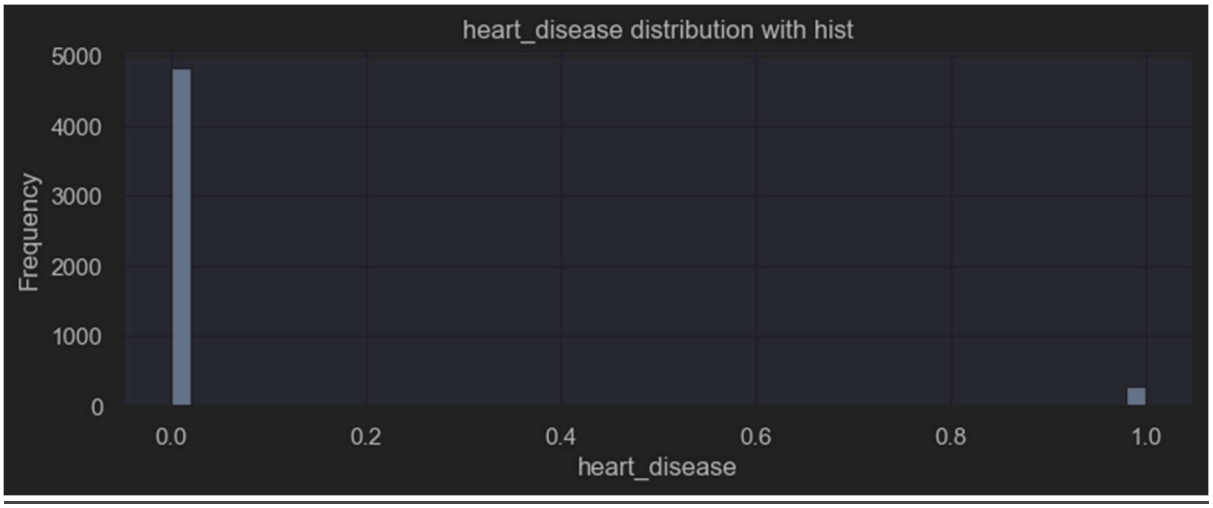

work_type
Frequency
3000
2500
2000
1500
1000
500
0
Private
Self-employed
children
Govt_job
Never_worked

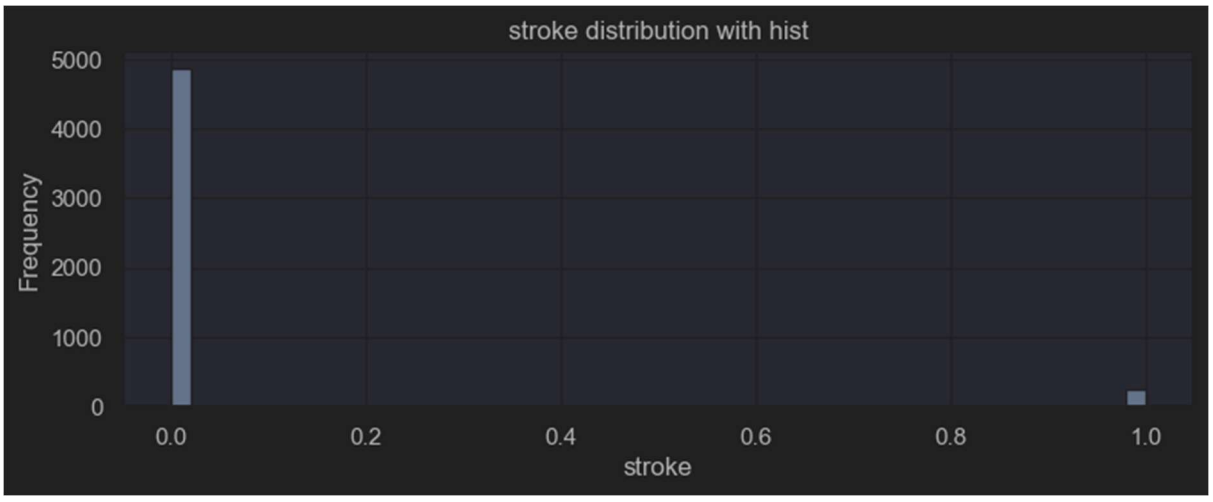

Residence_type
Frequency
2500
2000
1500
1000
500
0
Urban
Rural

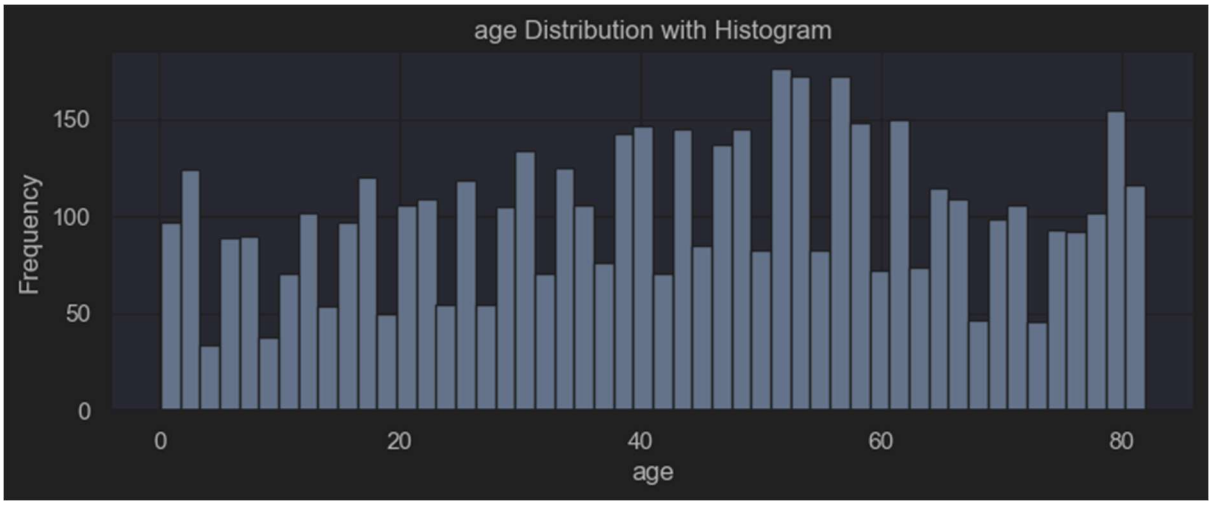

smoking_status
Frequency
1500
1000
500
0
never smoked
Unknown
formerly smoked
smokes

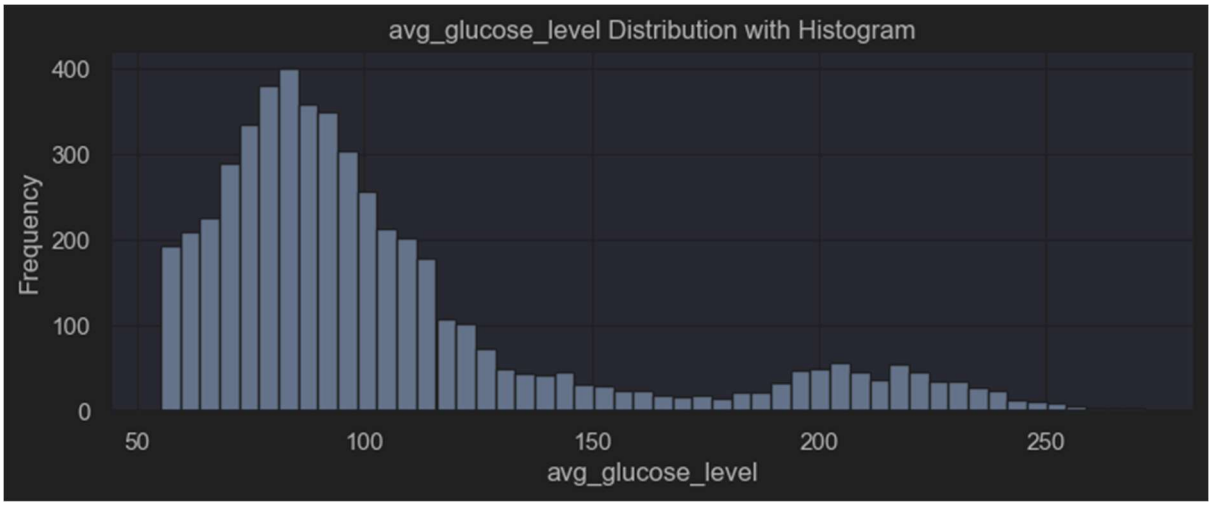

stroke
1.0
0.8
0.6
0.4
0.2
0.0
Yes
No
ever_married

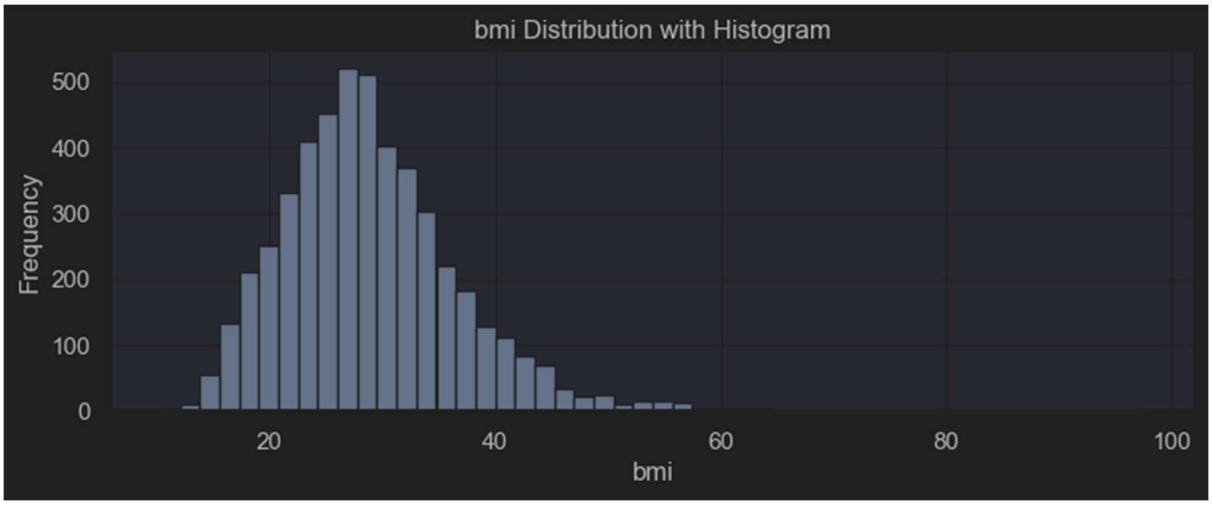

hypertension distribution with hist
Frequency
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
hypertension

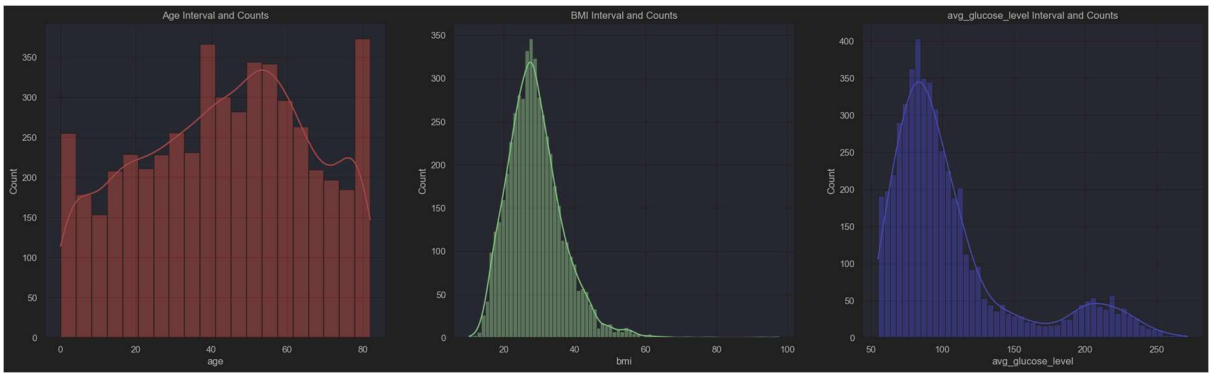

heart_disease distribution with hist
Frequency
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
heart_disease

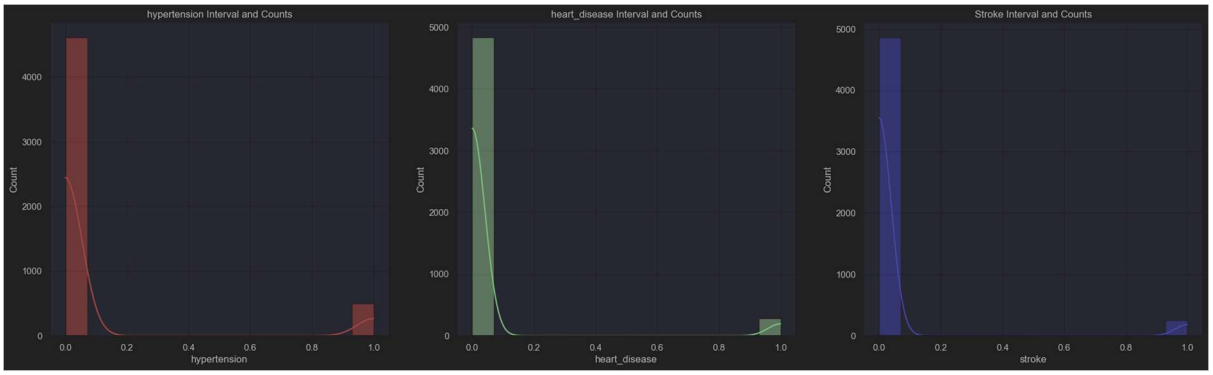

stroke distribution with hist
Frequency
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
stroke

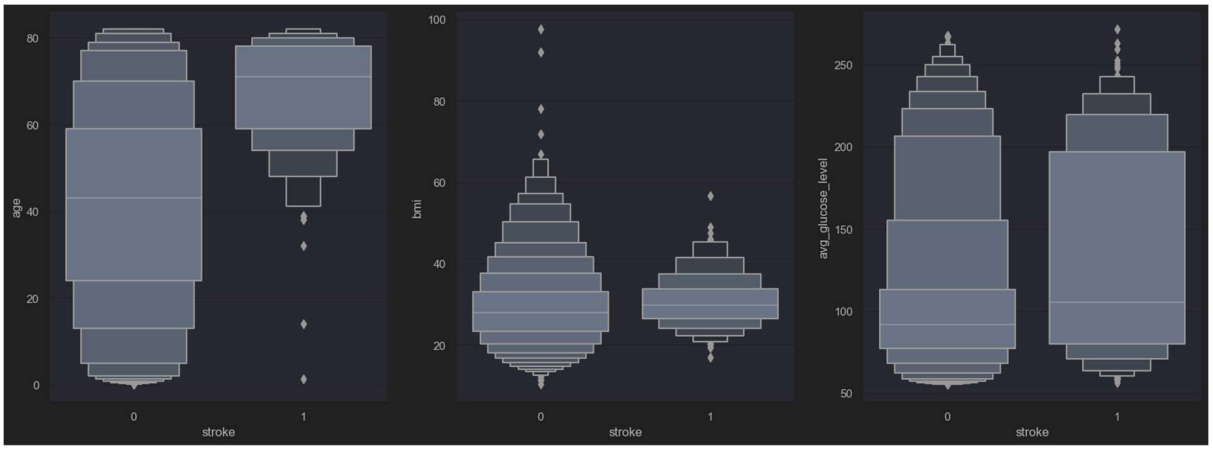

age Distribution with Histogram
Frequency
150
100
50
0
0
20
40
60
80
age

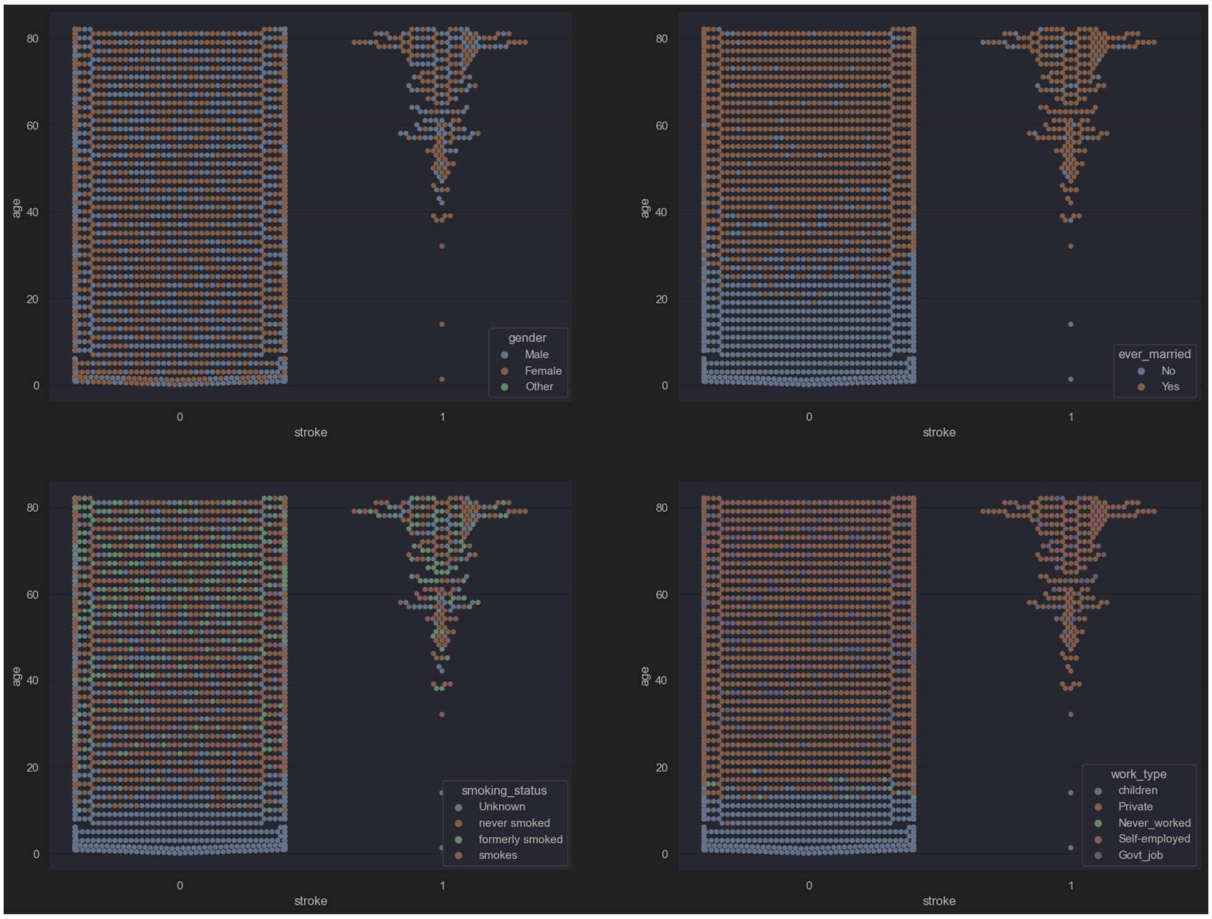

avg_glucose_level Distribution with Histogram
Frequency
400
300
200
100
0
50
100
150
200
250
avg_glucose_level


bmi Distribution with Histogram
Frequency
500
400
300
200
100
0
20
40
60
80
100
bmi

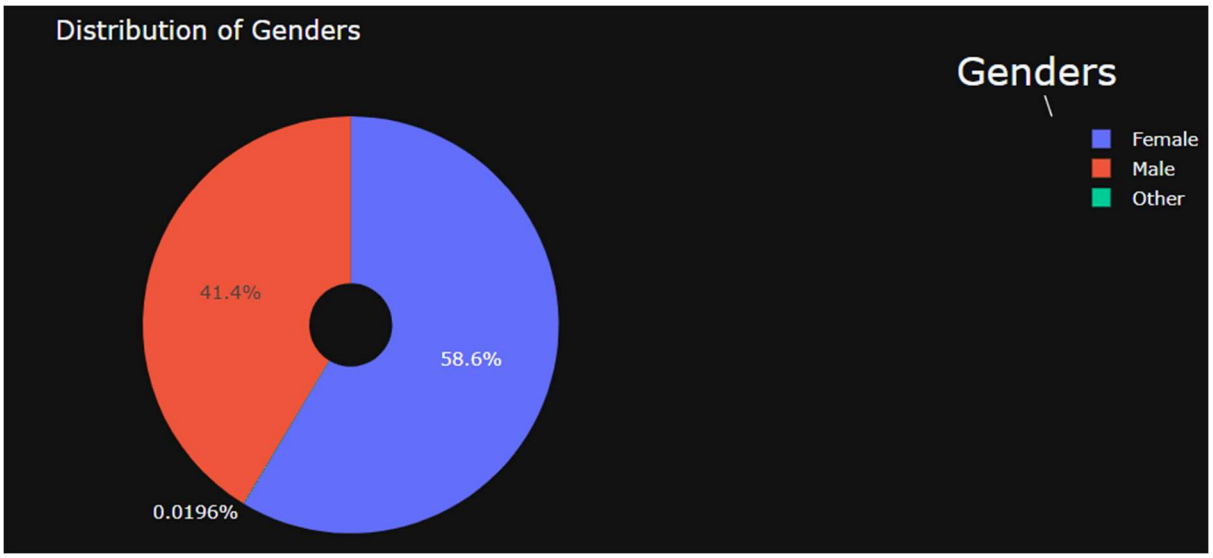

Age Interval and Counts
Count
age
BMI Interval and Counts
Count
bmi
avg_glucose_level Interval and Counts
Count
avg_glucose_level

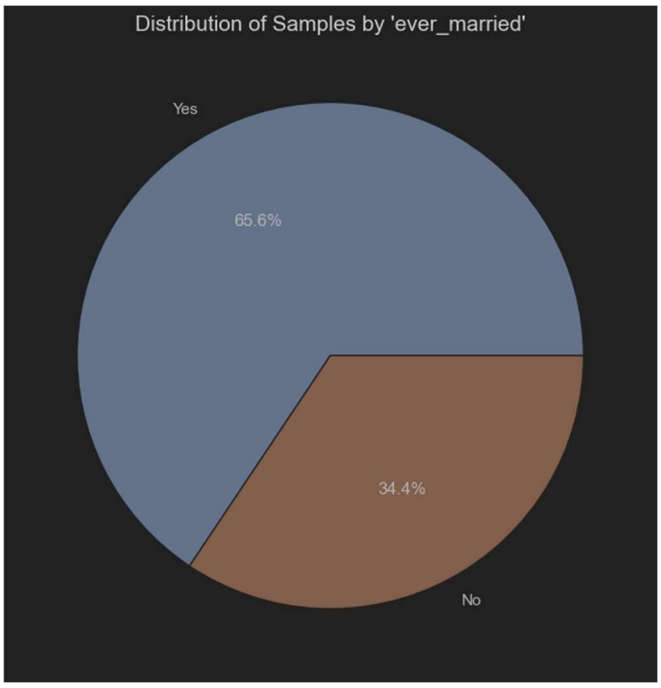

hypertension Interval and Counts
Count
hypertension
heart_disease Interval and Counts
Count
heart_disease
Stroke Interval and Counts
Count
stroke

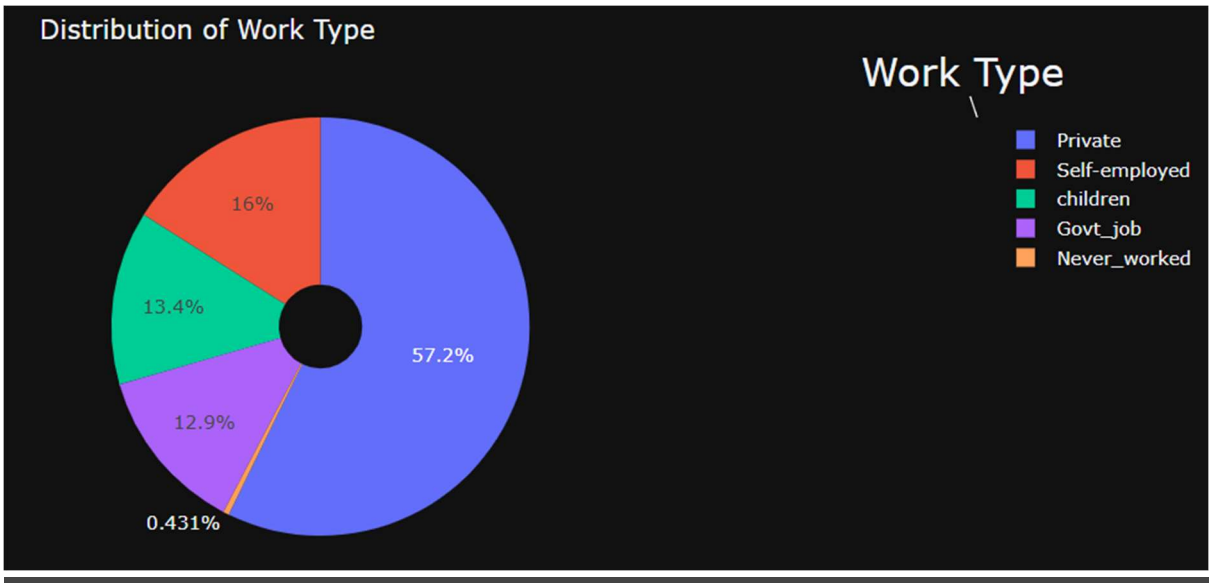

age
stroke
bmi
stroke
avg_glucose_level
stroke

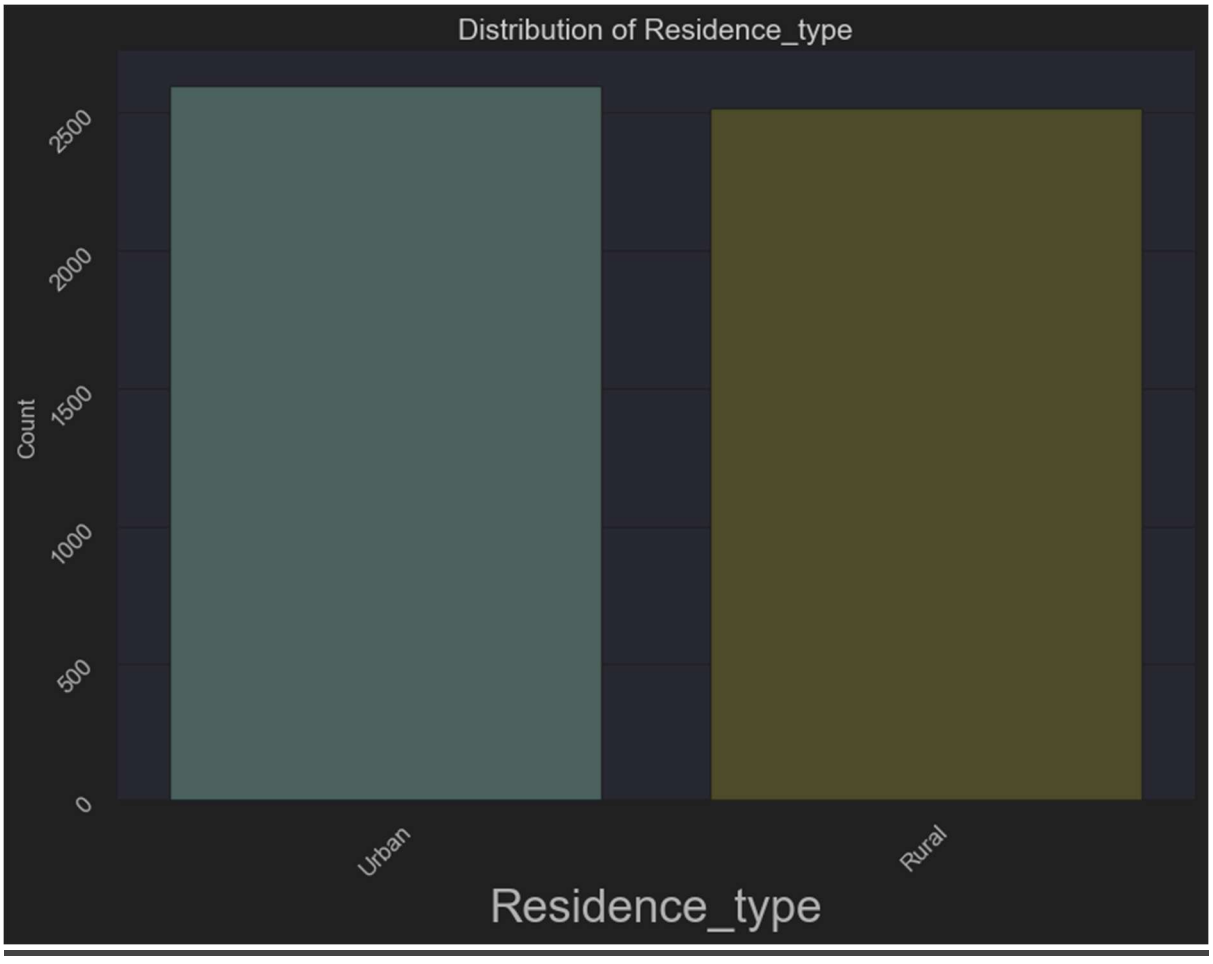

age
stroke
gender
Male
Female
Other
ever_married
No
Yes
smoking_status
Unknown
never smoked
formerly smoked
smokes
work_type
children
Private
Never_worked
Self-employed
Govt_job

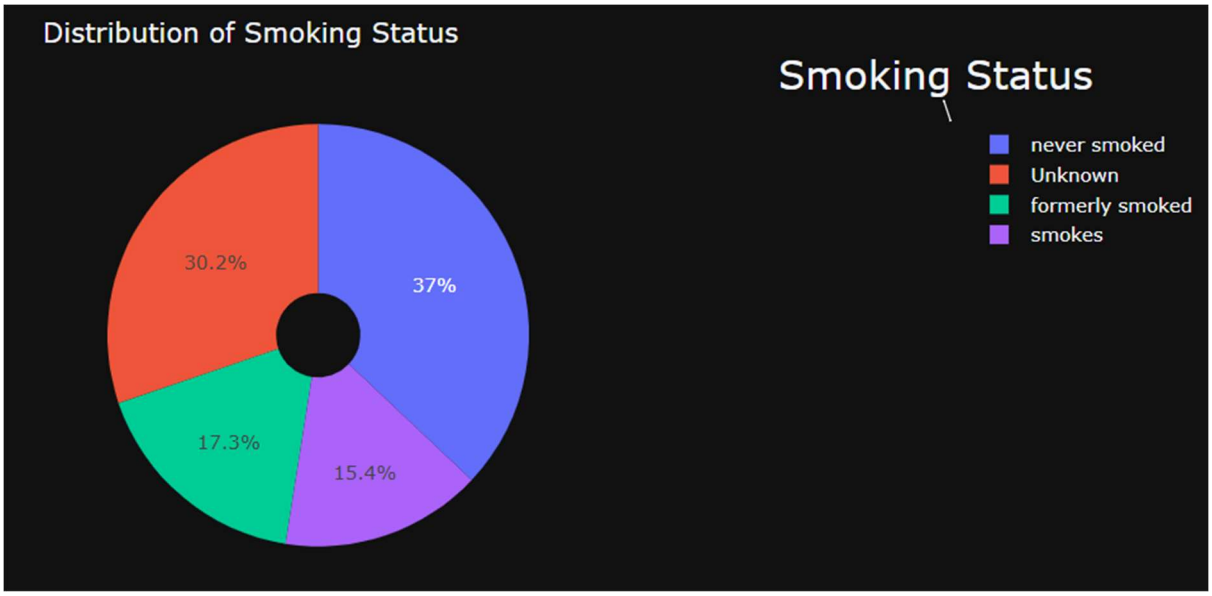
Basic Data Analysis





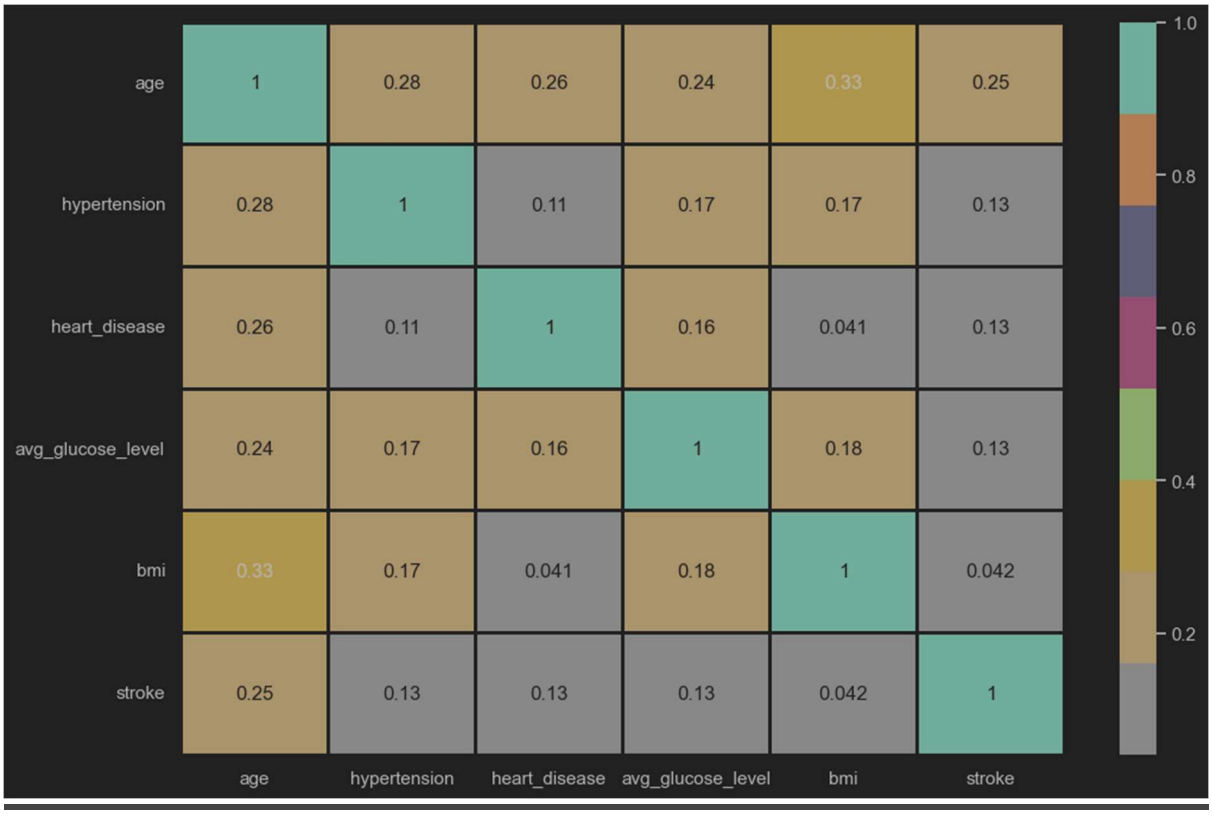

Distribution of Work Type
Work Type
Private
Self-employed
children
Govt_job
Never_worked
16%
13.4%
57.2%
12.9%
0.431%

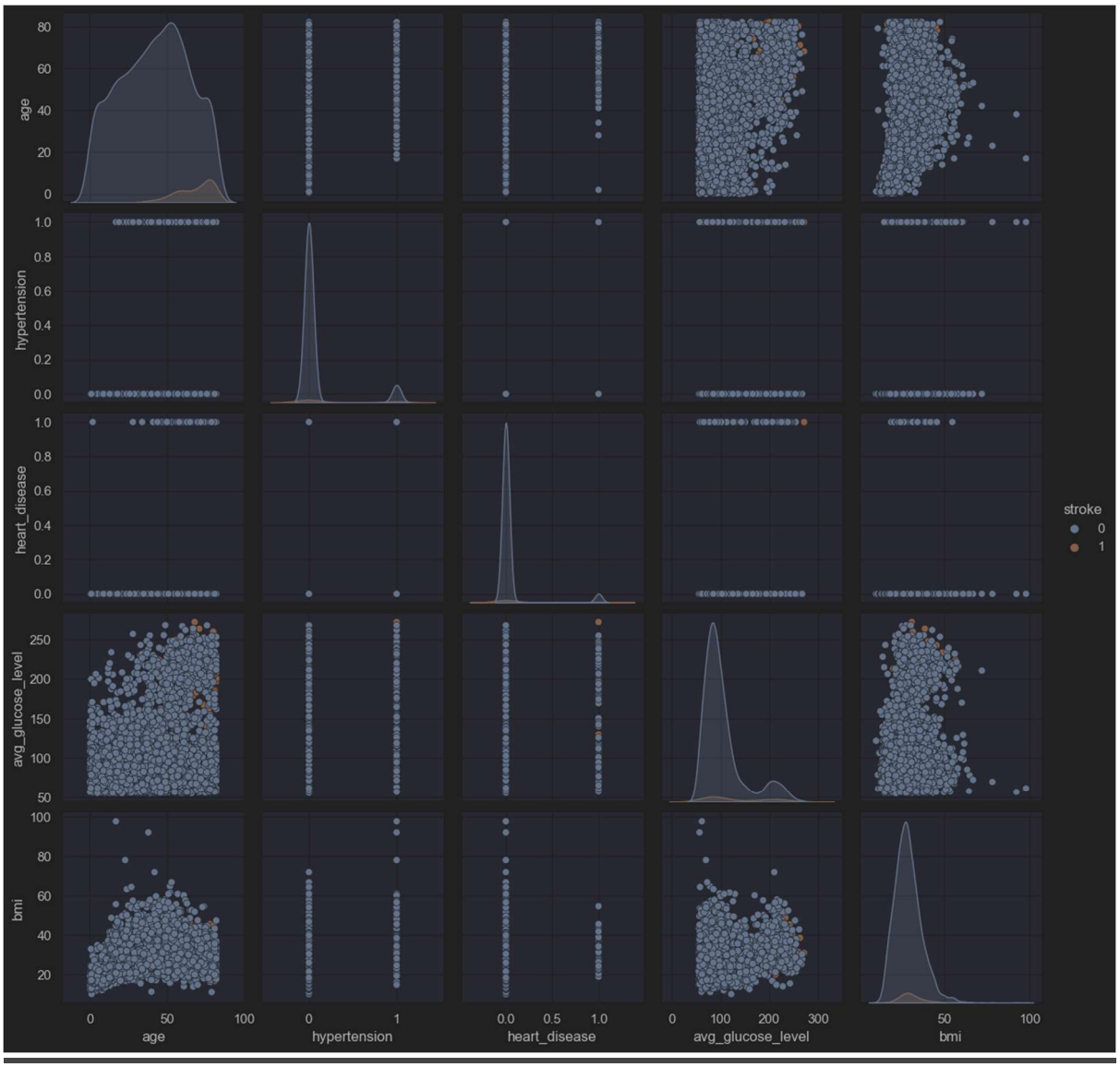

Distribution of Residence_type
Count
2500
2000
1500
1000
500
0
Urban
Rural
Residence_type



3. Correlation

Correlation explains how one or more variables are related to each other. These variables can be input data features which have been used to forecast our target variable.




age
hypertension
heart_disease
avg_glucose_level
bmi
stroke
0
1

4. Anomaly Detection

Anomaly is one that differs / deviates significantly from other observations in the same sample. An anomaly detection pattern produces two different results. The first is a categorical tag for whether the observation is abnormal or not; the second is a score or trust value. Score carries more information than the label. Because it also tells us how abnormal the observation is. The tag just tells you if it's abnormal. While labeling is more common in supervised methods, the score is more common in unsupervised and semisupervised methods.

```
def detect_outliers(df,features):
    outlier_indices = []

    for c in features:
        # 1st quartile
        Q1 = np.percentile(df[c],25)
        # 3rd quartile
        Q3 = np.percentile(df[c],75)
        # IQR
        IQR = Q3 - Q1
        # Outlier step
        outlier_step = IQR * 1.5
        # detect outlier and their indeces
        outlier_list_col = df[(df[c] < Q1 - outlier_step) | (df[c] > Q3 + outlier_step)].index
        # store indeces
        outlier_indices.extend(outlier_list_col)

    outlier_indices = Counter(outlier_indices)
    multiple_outliers = list(i for i, v in outlier_indices.items() if v > 2)

    return multiple_outliers
```

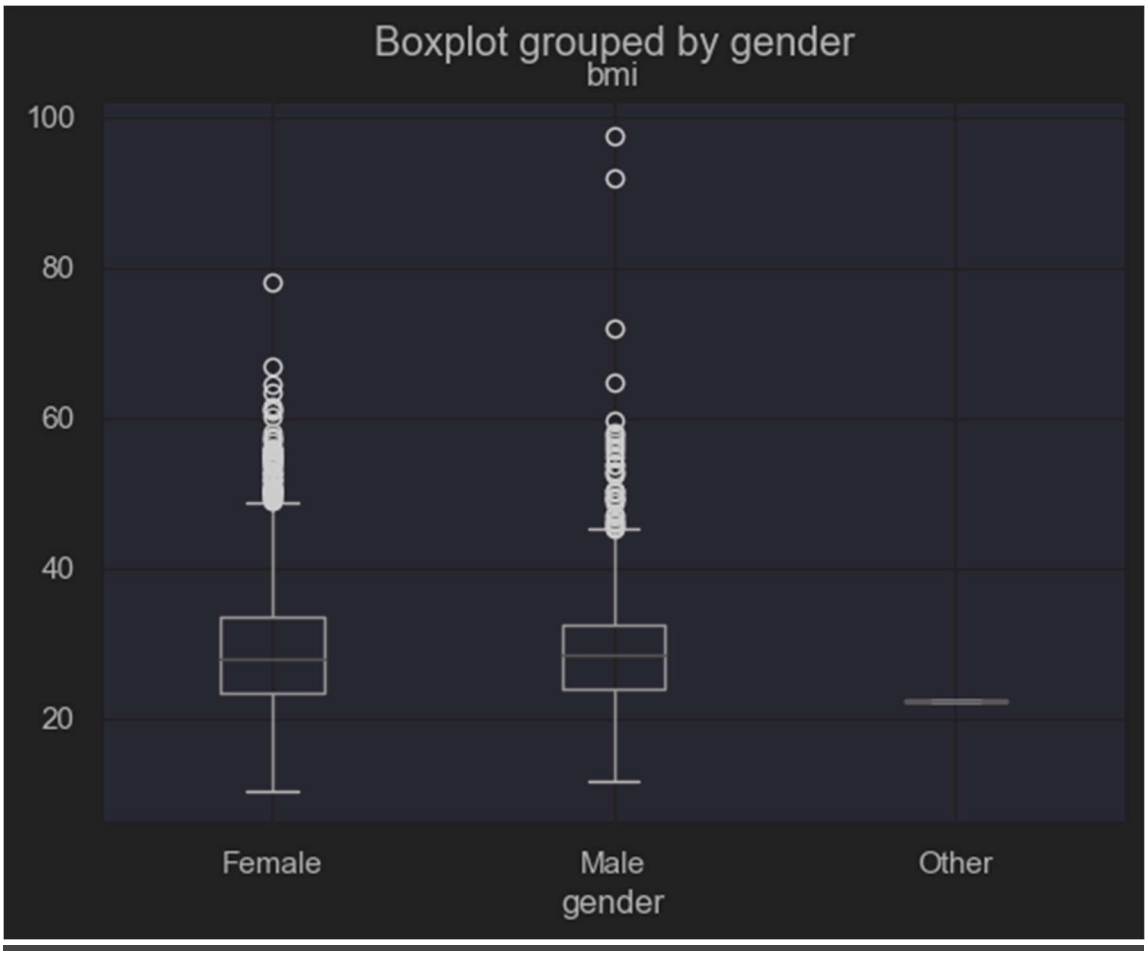
75 anomalies were detected and were dropped off the dataset.

5. Missing Values

We have 201 null values in total. bmi includes all. (After Anomaly Detection, it decreases to 192)

How can we handle null values?

- We have selected the differences for BMI will be between gender.




```
for i in range(0,5035):
    if(dataset['bmi'][i] == 0):
        if(dataset['gender'][i] == 'Male'):
            dataset['bmi'][i] = 28.594683544303823
        elif(dataset['gender'][i] == 'Female'):
            dataset['bmi'][i] = 29.035926055109936
        else:
            dataset['bmi'][i] = 28.854652338161664
```

6. Encoding

Add Code Cell | Add Markdown

First, we will handle Categorical Values.

```
from sklearn.preprocessing import StandardScaler, LabelEncoder, OneHotEncoder
print("Unique Values for Gender", dataset['gender'].unique())
print("Unique Values for ever_married", dataset['ever_married'].unique())
print("Unique Values for work_type", dataset['work_type'].unique())
print("Unique Values for Residence_type", dataset['Residence_type'].unique())
print("Unique Values for smoking_status", dataset['smoking_status'].unique())
```

```
Unique Values for Gender ['Female' 'Male' 'Other']
Unique Values for ever_married ['Yes' 'No']
Unique Values for work_type ['Self-employed' 'Private' 'Govt_job' 'children' 'Never_worked']
Unique Values for Residence_type ['Rural' 'Urban']
Unique Values for smoking_status ['never smoked' 'smokes' 'formerly smoked' 'Unknown']
```

Label Encoding

Add Code Cell | Add Markdown Cell

Label Encoding is an encoding technique for handling categorical variables. In this technique, each data is assigned a unique integer.

One-Hot Encoding

One Hot Encoding is the binary representation of categorical variables. This process requires categorical values to be mapped to integer values first. Next, each integer value is represented as a binary vector with all values zero except the integer index marked with 1.

One Hot Encoding makes the representation of categorical data more expressive and easy. Many machine learning algorithms cannot work directly with categorical data, so categories must be converted to numbers. This operation is required for input and output variables that are categorical.

In this part, we converted categorical datas to the binary values. This operation increases the accuracy.

```
dataset = pd.concat([dataset, datasetDummies_gender], axis=1)
dataset = pd.concat([dataset, datasetDummies_work_type], axis=1)
dataset = pd.concat([dataset, datasetDummies_smoking_status], axis=1)
dataset
```

```
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 11 columns):
```

Before

```
5035 rows × 20 columns
```

After

7. Train - Test Split

```
features = ['age',
 'hypertension',
 'heart_disease',
 'ever_married',
 'Residence_type',
 'avg_glucose_level',
 'bmi',
 'gender_encoded_Female',
 'gender_encoded_Male',
 'gender_encoded_Other',
 'work_type_encoded_Govt_job',
 'work_type_encoded_Never_worked',
 'work_type_encoded_Private',
 'work_type_encoded_Self-employed',
 'work_type_encoded_children',
 'smoking_status_encoded_Unknown',
 'smoking_status_encoded_formerly smoked',
 'smoking_status_encoded_never smoked',
 'smoking_status_encoded_smokes']

label = ['stroke']

X = dataset[features]
y = dataset[label]
```

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=3)
# X_valid, X_test, y_valid, y_test = train_test_split(X_test, y_test, test_size=0.5, random_state=2)

print(f'Total # of sample in whole dataset: {len(X)}')
print(f'Total # of sample in train dataset: {len(X_train)}')
# print(f'Total # of sample in validation dataset: {len(X_valid)}')
print(f'Total # of sample in test dataset: {len(X_test)}')
```

```
Total # of sample in whole dataset: 5035
Total # of sample in train dataset: 4028
Total # of sample in test dataset: 1007
```

Training Models and Test Scores

```
GaussianNB
Train score of trained model: 18.123138033763656
Test score of trained model: 17.477656405163852

Confussion Matrix:
[[145   0]
 [831  31]]

Accuracy : 0.17477656405163852
Specificity : 1.0

Classification Report:
              precision    recall  f1-score   support

           0       0.15      1.00      0.26       145
           1       1.00      0.04      0.07       862

    accuracy                           0.17      1007
   macro avg       0.57      0.52      0.16      1007
weighted avg       0.88      0.17      0.10      1007
```

```
BernoulliNB
Train score of trained model: 95.40714995034757
Test score of trained model: 95.72989076464746

Confussion Matrix:
[[964  31]
 [ 12   0]]

Accuracy : 0.9572989076464746
Specificity : 0.9688442211055276

Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       995
           1       0.00      0.00      0.00        12

    accuracy                           0.96      1007
   macro avg       0.49      0.48      0.49      1007
weighted avg       0.98      0.96      0.97      1007
```

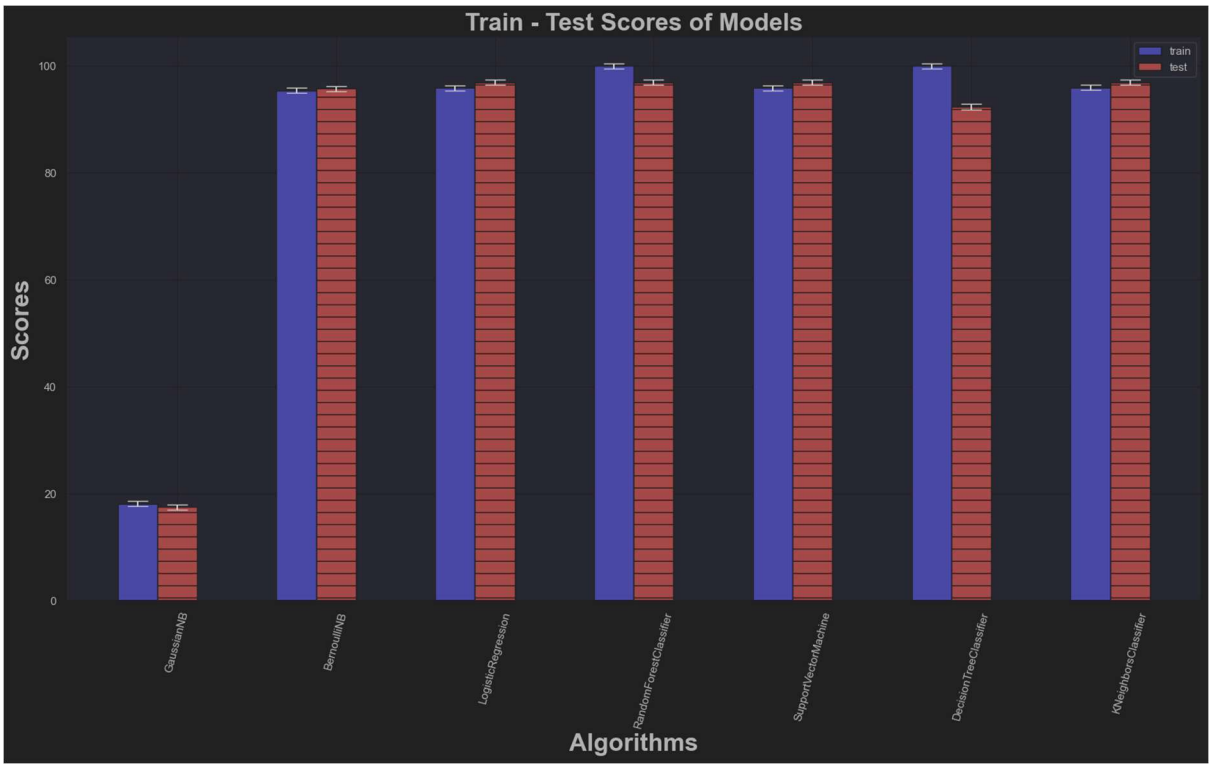
```
LogisticRegression
Train score of trained model: 95.87884806355511
Test score of trained model: 96.92154915590864

Confussion Matrix:
[[976  31]
 [  0   0]]

Accuracy : 0.9692154915590864
Specificity : 0.9692154915590864

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98      1007
           1       0.00      0.00      0.00         0

    accuracy                           0.97      1007
   macro avg       0.50      0.48      0.49      1007
weighted avg       1.00      0.97      0.98      1007
```

```
RandomForestClassifier
Train score of trained model: 100.0
Test score of trained model: 96.92154915590864

Confussion Matrix:
[[975  30]
 [  1   1]]

Accuracy : 0.9692154915590864
Specificity : 0.9701492537313433

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98      1005
           1       0.03      0.50      0.06         2

    accuracy                           0.97      1007
   macro avg       0.52      0.74      0.52      1007
weighted avg       1.00      0.97      0.98      1007
```

```
SupportVectorMachine
Train score of trained model: 95.87884806355511
Test score of trained model: 96.92154915590864

Confussion Matrix:
[[976  31]
 [  0   0]]

Accuracy : 0.9692154915590864
Specificity : 0.9692154915590864

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98      1007
           1       0.00      0.00      0.00         0

    accuracy                           0.97      1007
   macro avg       0.50      0.48      0.49      1007
weighted avg       1.00      0.97      0.98      1007
```

```
DecisionTreeClassifier
Train score of trained model: 100.0
Test score of trained model: 92.35352532274081

Confussion Matrix:
[[924  25]
 [ 52   6]]

Accuracy : 0.9235352532274081
Specificity : 0.9736564805057956

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.97      0.96       949
           1       0.19      0.10      0.13        58

    accuracy                           0.92      1007
   macro avg       0.57      0.54      0.55      1007
weighted avg       0.90      0.92      0.91      1007
```

```
KNeighborsClassifier
Train score of trained model: 95.95332671300893
Test score of trained model: 96.92154915590864

Confussion Matrix:
[[976  31]
 [  0   0]]

Accuracy : 0.9692154915590864
Specificity : 0.9692154915590864

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98      1007
           1       0.00      0.00      0.00         0

    accuracy                           0.97      1007
   macro avg       0.50      0.48      0.49      1007
weighted avg       1.00      0.97      0.98      1007
```




```
Accuracy of GaussianNB -----> 17.477656405163852
Accuracy of BernoulliNB -----> 95.72989076464746
Accuracy of LogisticRegression -----> 96.92154915590864
Accuracy of RandomForestClassifier -----> 96.92154915590864
Accuracy of SupportVectorMachine -----> 96.92154915590864
Accuracy of DecisionTreeClassifier -----> 92.35352532274081
Accuracy of KNeighborsClassifier -----> 96.92154915590864
```

The Highest is the *"Accuracy of KNeighborsClassifier ——> 96.22641509433963"*

Comparing Models

We have used several measures to measure the significance of use of the models and then compare them using it.

- Accuracy is the percentage of correctly classified tweets out of the total number of tweets in the testing set.
- Precision measures the proportion of correctly classified positive tweets out of all tweets classified as positive.
- Recall measures the proportion of correctly classified positive tweets out of all actual positive tweets in the testing set.
- F1-score is the harmonic mean of precision and recall and is a more balanced measure of classifier performance.

We have also used a confusion matrix to get a better image of how well our models perform.

We got the following numbers for accuracy of different models:

- Accuracy of GaussianNB -----> 18.46031746031746
- Accuracy of BernoulliNB -----> 90.72222222222223
- Accuracy of LogisticRegression -----> 91.8555555555556
- Accuracy of RandomForestClassifier -----> 93.63492063492063
- Accuracy of SupportVectorMachine -----> 92.46031746031746
- Accuracy of DecisionTreeClassifier -----> 92.06349206349206
- Accuracy of KNeighborsClassifier -----> 96.03174603174604

Result

The aim of this project was to predict the occurrence of brain stroke using machine learning algorithms. The dataset was obtained from Kaggle, which contained information such as age, gender, hypertension, heart disease, smoking status, and various other factors.

The data was first preprocessed by performing exploratory data analysis and dealing with missing values and categorical variables. After preprocessing, the data was split into 80:20 train-test ratio. Seven machine learning algorithms, namely Gaussian Naïve Bayes, Bernoulli Naïve Bayes, Logistic Regression, Random Forest Classifier, Decision Tree Classifier, K Neighbors Classifier, and Support Vector Machine, were trained on the training data.

The accuracy of each algorithm was evaluated using various parameters, and the main focus was on accuracy. The results of the analysis showed that the K Neighbors Classifier algorithm had the highest accuracy with a score of 96.03%, followed by Random Forest Classifier (93.63%), Logistic Regression (91.86%), Support Vector Machine (92.46%), Decision Tree Classifier (92.06%), Bernoulli Naïve Bayes (90.72%), and Gaussian Naïve Bayes (18.46%).

In the future these models can be made more robust by training and testing against an even larger dataset. Further the best model can be developed into a portable application that can be easily accessed by an individual to warn them against a potential stroke.

Checklist for Dissertation-III Supervisor

Name: _____ UID: _____ Domain: _____

Registration No: _____Name of student:_____

Title of Dissertation:

_____

- ☐ Front pages are as per the format.
- ☐ Topic on the PAC form and title page are same.
- ☐ Front page numbers are in roman and for report, it is like 1, 2, 3…….
- ☐ TOC, List of Figures, etc. are matching with the actual page numbers in the report.
- ☐ Font, Font Size, Margins, line Spacing, Alignment, etc. are as per the guidelines.
- ☐ Color prints are used for images and implementation snapshots.
- ☐ Captions and citations are provided for all the figures, tables etc. and are numbered and center aligned.
- ☐ All the equations used in the report are numbered.
- ☐ Citations are provided for all the references.
- ☐ **Objectives are clearly defined.**
- ☐ Minimum total number of pages of report is 50.
- ☐ Minimum references in report are 30.

Here by, I declare that I had verified the above mentioned points in the final dissertation report.

Signature of Supervisor with UID