Evaluation of Machine Learning Models for Predicting Brain Stroke

A Dissertation

Submitted in partial fulfilment for the degree of

Master of Arts

In

Biostatistics

Submitted by

PRAKHAR KUMAR

Roll Number: 2210011235004

Under the guidance of

Dr. ASHOK KUMAR

Department of statistics, University of Lucknow

Year 2024

Evaluation of Machine Learning Models for Predicting Brain Stroke

Submitted to:

Department of Statistics

University of Lucknow

Under the supervision of:	**Submitted by:**
Dr. Ashok Kumar	Prakhar Kumar
Department of Statistics	M.A. Biostatistics
University of Lucknow	IV Semester

VIVA-VOCA CERTIFICATE

This is to certify that the thesis entitled " **Evaluation of Machine Learning Models for Predicting Brain Stroke**" which is being submitted by Mr. Prakhar Kumar under the supervision of Dr. Ashok Kumar, Professor, Department of Statistics, University of Lucknow, Lucknow, for the award of M.A.(Biostatistics) has been approved after an oral examination of the same in collaboration with an external examiner

External Examiner	Internal Examiner	Head
(Name & Signature)	(Name & Signature)	(Name & Signature)

Date:

Place: Department of Statistics, University of Lucknow, Lucknow

CERTIFICATE OF SUPERVISOR

This is to certify that Mr. Prakhar Kumar has completed the necessary academic term and the work presented by his is an original record of his own work carried out by him/her under the supervision of Dr. Ashok Kumar, Department of Statistics, University of Lucknow, Lucknow. He has worked "**Evaluation of Machine Learning Models for Predicting Brain Stroke**". To the best of my knowledge, the matter embodied in this thesis has not been submitted to any other University or Institution for the award of any degree. In my opinion, it is worthy of consideration for the award of M.A. Degree in 'Bio-Statistics' to the candidate.

Supervisor

Dr. Ashok Kumar

Department of StatisticsUniversity of Lucknow, Lucknow

CANDIDATE CERTIFICATE

I hereby declare that this dissertation entitled "**Evaluation of Machine Learning Models for Predicting Brain Stroke**" carried out by me under the supervision of Dr. Ashok Kumar, Department of Statistics, University of Lucknow, Lucknow. It is an original work and does not contain any work submitted for the award of degree in this university of any other University or Institution.

Date: Prakhar Kumar

Place: Department of Statistics M.Sc.(Bio-Statistics)

University of Lucknow Roll No. 2210011235004

ACKNOWLDGEMENT

I would like to express my deepest gratitude to all those who have supported and guided me throughout the completion of this thesis on "**Evaluation of Machine Learning Models for Predicting Brain Stroke** ". First and foremost, I would like to thank my thesis supervisor Dr. Ashok Kumar, University of Lucknow, Lucknow, for her invaluable guidance, encouragement and insightful feedback throughout the research process. I express my sincere gratitude towards Prof. Masood H. Siddiqui, Head of Department of Statistics, University of Lucknow for allowing me to conduct my thesis and providing me with all the departmental facilities towards the completion of this thesis. A special thanks to the faculty and staff of the Department of Statistics, University of Lucknow for providing a stimulating academic environment and the necessary resources to conduct my research.

My sincere appreciation goes to my friends, for the continuous encouragement, insightful discussions and for making this journey enjoyable. Finally, I am deeply indebted to my family for their unwavering support and understanding. Their patience, love and encouragement have been my greatest motivation throughout this academic endeavour. Thank you all for your contributions and support.

CONTENTS

Chapter-1

Introduction

Stroke, a devastating neurological disorder, exacts a significant toll on global health, with millions falling victim to its debilitating effects each year. According to the World Stroke Organization, a staggering 13 million individuals are struck by stroke annually, resulting in approximately 5.5 million deaths. Its pervasive impact extends beyond mere mortality, profoundly affecting survivors, their families, social networks, and workplaces. Contrary to common misconceptions, stroke does not discriminate based on age, gender, or physical condition, underscoring the urgency for comprehensive prevention and management strategies.

The clinical presentation of stroke is diverse, encompassing a spectrum of motor, sensory, and cognitive deficits. Symptoms can manifest abruptly or evolve gradually, with some individuals experiencing nocturnal onset. Paralysis, numbness, speech impairment, vertigo, and headache typify common stroke symptoms, underscoring the heterogeneous nature of its clinical phenotype. Prompt diagnosis, facilitated by neuroimaging modalities such as computed tomography (CT) and magnetic resonance imaging (MRI), is imperative for timely intervention and prognostication.

Stroke, often referred to as a "brain attack," is a medical emergency that occurs when blood flow to a part of the brain is interrupted or reduced, leading to oxygen deprivation and damage to brain cells. This disruption in blood flow can result from either a blockage in the blood vessels (ischemic stroke) or bleeding into the brain (hemorrhagic stroke). Stroke is a leading cause of disability and death worldwide, emphasizing the importance of understanding its types, symptoms,` and treatment options.

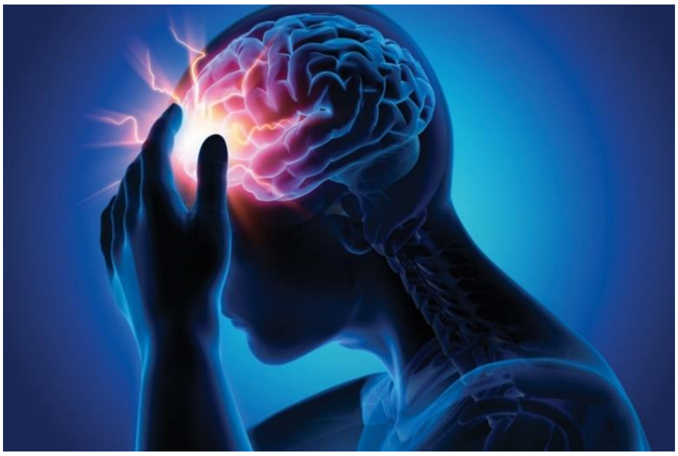
History of Brain Stroke:

Episodes of stroke and familial stroke have been reported from the 2nd millennium BC onward in ancient Mesopotamia and Persia. Hippocrates (460 to 370 BC) was first to describe the phenomenon of sudden paralysis that is often associated with ischemia. Apoplexy, from the Greek word meaning "struck down with violence", first appeared in Hippocratic writings to describe this phenomenon. The word stroke was used as a synonym for apoplectic seizure as early as 1599, and is a fairly literal translation of the Greek term. The term apoplectic stroke is an archaic, nonspecific term, for a cerebrovascular accident accompanied by haemorrhage or haemorrhagic stroke. Martin Luther was described as having an apoplectic stroke that deprived him of his speech shortly before his death in 1546.

In 1658, in his Apoplexia, Johann Jacob Wepfer (1620–1695) identified the cause of haemorrhagic stroke when he suggested that people who had died of apoplexy had bleeding in their brains. Wepfer also identified the main arteries supplying the brain, the vertebral and carotid arteries, and identified the cause of a type of ischemic stroke known as a cerebral infarction when he suggested that apoplexy might be caused by a blockage to those vessels. Rudolf Virchow first described the mechanism of thromboembolism as a major factor.

The term cerebrovascular accident was introduced in 1927, reflecting a "growing awareness and acceptance of vascular theories and recognition of the consequences of a sudden disruption in the vascular supply of the brain". Its use is now discouraged by a number of neurology textbooks, reasoning that the connotation of fortuitousness carried by the word accident insufficiently highlights the modifiability of the underlying risk factors. Cerebrovascular insult may be used interchangeably.

The term brain attack was introduced for use to underline the acute nature of stroke according to the American Stroke Association, which has used the term since 1990, and is used colloquially to refer to both ischemic as well as haemorrhagic stroke.



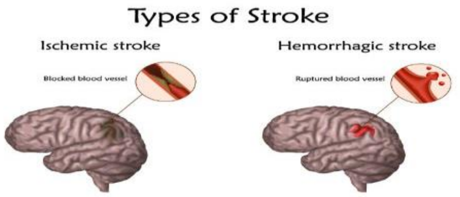
Types of Strokes:

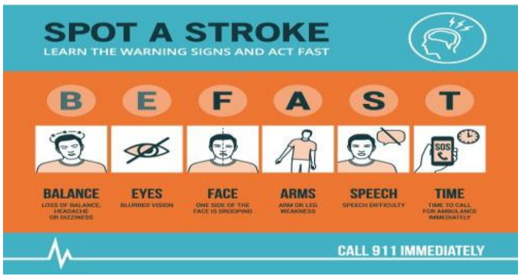
1. Ischemic Stroke:

- Ischemic strokes account for approximately 87% of all stroke cases and occur when a blood clot or plaque buildup blocks or narrows an artery supplying blood to the brain.
- The two main subtypes of ischemic stroke include thrombotic strokes, which occur due to the formation of a blood clot within an artery supplying the brain, and embolic strokes, which result from a blood clot that forms elsewhere in the body and travels to the brain.
- Risk factors for ischemic stroke include hypertension, diabetes, high cholesterol, smoking, and obesity.

2. **Hemorrhagic Stroke:**

- Hemorrhagic strokes occur when a weakened blood vessel ruptures and bleeds into the surrounding brain tissue, causing damage and swelling.
- The two primary types of hemorrhagic stroke are intracerebral hemorrhage, which involves bleeding within the brain tissue, and subarachnoid hemorrhage, which occurs when blood accumulates in the space between the brain and the surrounding membranes.
- Risk factors for hemorrhagic stroke include hypertension, cerebral aneurysms, arteriovenous malformations (AVMs), and anticoagulant medications.

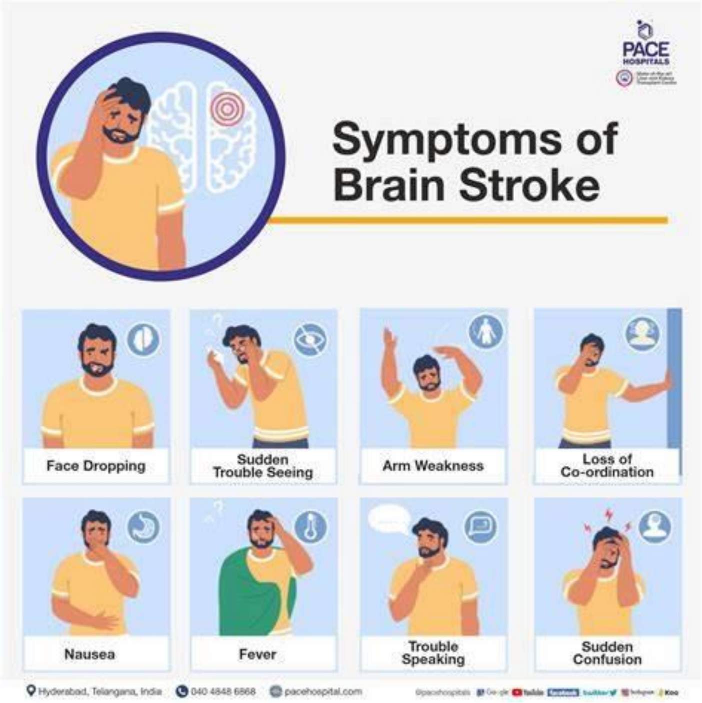



SPOT A STROKE
LEARN THE WARNING SIGNS AND ACT FAST
B
E
F
A
S
T
SOS
BALANCE
LOSS OF BALANCE, HEADACHE OR DIZZINESS
EYES
BLURRED VISION
FACE
ONE SIDE OF THE FACE IS DROOPING
ARMS
ARM OR LEG WEAKNESS
SPEECH
SPEECH DIFFICULTY
TIME
TIME TO CALL FOR AMBULANCE IMMEDIATELY
CALL 911 IMMEDIATELY

Symptoms of Stroke: The symptoms of a stroke can vary depending on the type of stroke, the severity of the brain damage, and the area of the brain affected. However, there are common signs and symptoms that may indicate a stroke is occurring:

1. Sudden numbness or weakness, especially on one side of the body, often affecting the face, arm, or leg.
2. Difficulty speaking or understanding speech (aphasia).
3. Confusion or trouble with comprehension.

4. Vision problems, such as blurred or double vision.
5. Severe headache with no apparent cause.
6. Trouble walking, dizziness, or loss of balance and coordination.
7. Sudden onset of severe headache. These symptoms typically come on suddenly and require immediate medical attention.



Transient Ischemic Attacks (TIAs)

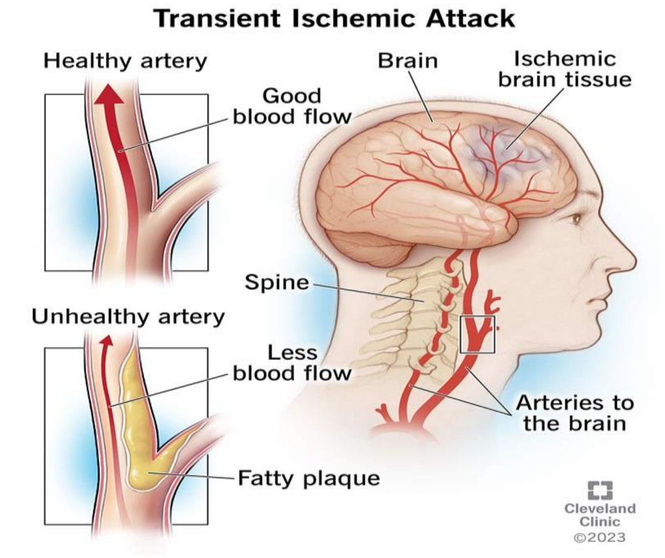
Transient ischemic attacks (TIAs), often referred to as "mini-strokes," produce stroke-like symptoms but typically resolve within a few minutes to hours without causing permanent damage. However, TIAs are warning signs of an impending stroke and should not be ignored. Individuals who experience a TIA are at an increased risk of having a full-blown stroke in the future and should seek medical evaluation and treatment to reduce this risk.

Symptoms:

Transient ischemic attacks usually last a few minutes. Most symptoms disappear within an hour. Rarely, symptoms may last up to 24 hours. The symptoms of a TIA are similar to those found early in a stroke. Symptoms happen suddenly and may include:

- Weakness, numbness or paralysis in the face, arm or leg, typically on one side of the body.
- Slurred speech or trouble understanding others.
- Blindness in one or both eyes or double vision.
- Dizziness or loss of balance or coordination.

You may have more than one TIA. Their symptoms may be similar or different depending on which area of the brain is involved.



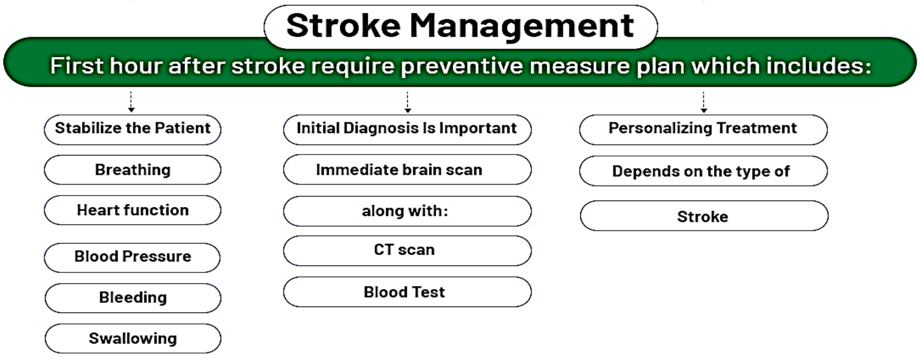
Diagnosis and Treatment: The diagnosis of stroke typically involves a combination of medical history, physical examination, imaging tests (such as CT scans or MRI), and blood tests. Once a stroke is diagnosed, prompt treatment is essential to minimize brain damage and improve outcomes. Treatment options for stroke include:

1. **Ischemic Stroke Treatment:**

- Intravenous thrombolysis: The administration of clot-busting medications, such as tissue plasminogen activator (TPA), within a specific time window after the onset of symptoms can help dissolve blood clots and restore blood flow to the brain.
- Mechanical thrombectomy: In cases of large vessel occlusion, a procedure called mechanical thrombectomy may be performed to remove the clot directly from the blocked artery using specialized devices.

2. **Hemorrhagic Stroke Treatment:**

- Controlling bleeding: In cases of hemorrhagic stroke, the primary goal of treatment is to stop the bleeding and reduce pressure on the brain. This may involve medications to lower blood pressure, surgical interventions to repair or remove damaged blood vessels, or minimally invasive procedures such as endovascular coiling or embolization.
- Supportive care: Patients with hemorrhagic stroke may require intensive care monitoring and supportive measures to manage complications such as cerebral edema (swelling of the brain) or seizures.



Stroke Management

First hour after stroke require preventive measure plan which includes:

Stabilize the Patient

Breathing

Heart function

Blood Pressure

Bleeding

Swallowing

Initial Diagnosis Is Important

Immediate brain scan

along with:

CT scan

Blood Test

Personalizing Treatment

Depends on the type of

Stroke

Rehabilitation and Recovery: Stroke rehabilitation plays a crucial role in helping individuals regain lost function, improve quality of life, and prevent complications. Depending on the extent of brain damage and the specific impairments experienced, stroke rehabilitation may involve physical therapy, occupational therapy, speech therapy, and other specialized interventions. The goal of rehabilitation is to maximize independence and facilitate a return to activities of daily living, work, and social participation.

While some risk factors for stroke, such as age and family history, cannot be modified, there are several lifestyle changes and medical interventions that can help reduce the risk of stroke:

1. **Manage hypertension:** High blood pressure is the single most important risk factor for stroke and should be monitored and managed through lifestyle modifications (such as a healthy diet, regular exercise, and stress management) and medications as needed.
2. **Control diabetes:** Individuals with diabetes should strive to maintain optimal blood sugar levels through diet, exercise, medication, and regular monitoring.
3. **Maintain a healthy diet:** A diet rich in fruits, vegetables, whole grains, and lean proteins, and low in saturated and trans fats, cholesterol, and sodium can help reduce the risk of stroke and other cardiovascular diseases.

4. **Be physically active:** Regular exercise can help improve cardiovascular health, lower blood pressure, reduce cholesterol levels, and maintain a healthy weight, all of which contribute to stroke prevention.

5. **Avoid smoking and excessive alcohol consumption**: Smoking and heavy alcohol consumption are significant risk factors for stroke and should be avoided or minimized.

6. **Treat atrial fibrillation:** Atrial fibrillation, a heart rhythm disorder characterized by irregular and rapid heartbeats, increases the risk of stroke and should be managed with medications or procedures to prevent blood clots.

Stroke rehabilitation often involves a team of people with various specializations.
Rehabilitation goals after a stroke include:
Medical doctor
Rehabilitation nurse
Speech therapist
Physical therapist
Occupational therapist
Improving speech and language skills and treating problems with swallowing
Improving mobility, balance, and walking, and preventing complications, such as contractures
Improving function in daily activities
K. BUCHER

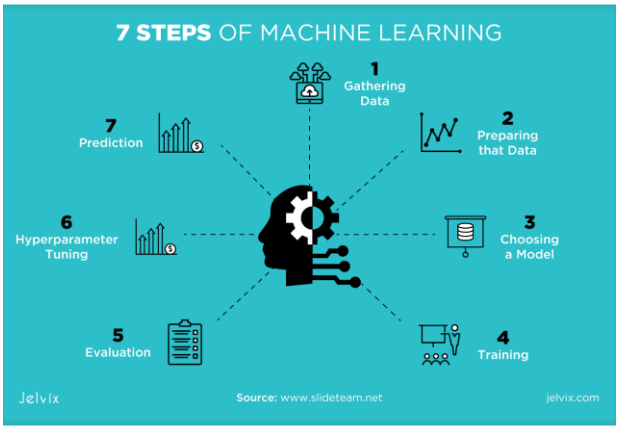
In recent years, the advent of information and communication technologies (ICTs), particularly artificial intelligence (AI) and machine learning (ML), has revolutionized healthcare delivery, offering novel avenues for disease prediction and management. Leveraging large-scale datasets and sophisticated algorithms, researchers have endeavoured to develop predictive models for various medical conditions, including stroke. Central to these efforts is the synthetic minority oversampling technique

(SMOTE), a data augmentation method aimed at mitigating class imbalance inherent in stroke datasets.

The burgeoning field of ML holds immense promise for stroke prediction, offering a repertoire of algorithms ranging from naive Bayes and logistic regression to ensemble methods like random forests and stacking. By harnessing the collective predictive power of diverse ML models, researchers have endeavoured to enhance the accuracy and generalizability of stroke prediction algorithms. Of particular note is the stacking method, which amalgamates predictions from multiple base learners, yielding superior performance metrics compared to standalone models.

Machine learning, a subset of artificial intelligence, has emerged as a powerful tool in modern technology, transforming the way we approach problems and make decisions. At its core, machine learning involves the development of algorithms and models that enable computers to learn from data and improve their performance over time without being explicitly programmed.

One of the fundamental concepts in machine learning is the utilization of algorithms to analyze data and extract meaningful patterns or insights. These algorithms are designed to learn from past experiences or observations and use that knowledge to make predictions or decisions about new data. This process is akin to how humans learn from experience; however, machine learning algorithms can process vast amounts of data at incredible speeds, enabling them to uncover complex patterns that may not be immediately apparent to humans.



There are several broad categories of machine learning algorithms, each with its unique characteristics and applications. Supervised learning is one of the most common approaches, where the algorithm is trained on labeled data, meaning that each input data point is associated with a corresponding output label. The algorithm learns to map inputs to outputs by identifying patterns and relationships in the training data. Once trained, the model can then make predictions on new, unseen data based on its learned associations.

In conclusion, stroke represents a formidable public health challenge, necessitating a multifaceted approach encompassing prevention, acute management, and rehabilitation. The integration of AI and ML holds tremendous potential for advancing stroke care, enabling early risk stratification, personalized treatment algorithms, and targeted rehabilitation interventions. Moving forward, collaborative efforts between clinicians, researchers, and technologists are paramount to harnessing the transformative power of ML in combating the scourge of stroke.

LITERATURE REVIEW

Sirsat MS et al., 2020 conducted the review in which Machine Learning (ML) delivers an accurate and quick prediction outcome and it has become a powerful tool in health settings, offering personalized clinical care for stroke patients. An application of ML and Deep Learning in health care is growing however, some research areas do not catch enough attention for scientific investigation though there is real need of research. Therefore, the aim of this work is to classify state-of-arts on ML techniques for brain stroke into 4 categories based on their functionalities or similarity, and then review studies of each category systematically. A total of 39 studies were identified from the results of ScienceDirect web scientific database on ML for brain stroke from the year 2007 to 2019. Support Vector Machine (SVM) is obtained as optimal models in 10 studies for stroke problems. Besides, maximum studies are found in stroke diagnosis although number for stroke treatment is least thus, it identifies a research gap for further investigation. Similarly, CT images are a frequently used dataset in stroke. Finally SVM and Random Forests are efficient techniques used under each category. They present study showcases the contribution of various ML approaches applied to brain stroke.

Dritsas E et al., 2022 conducted a study for stroke risk rediction with machine learning technique Abstract: A stroke is caused when blood flow to a part of the brain is stopped abruptly. Without the blood supply, the brain cells gradually die, and disability occurs depending on the area of the brain affected. Early recognition of symptoms can significantly carry valuable information for the prediction of stroke and promoting a healthy life. In this research work, with the aid of machine learning (ML), several models are developed and evaluated to design a robust framework for the long- term risk prediction of stroke occurrence. The main contribution of this study is a stacking method that achieves a high performance that is validated by various metrics, such as AUC, precision, recall, F-measure and accuracy. Their results showed that the stacking classification outperforms the other methods, with an AUC of 98.9%, F-measure, precision and recall of 97.4% and an accuracy of 98%.

Rahman S et al., 2023 proposed a the study on Prediction of brain stroke using machine learning algorithms and deep neural network techniques. The brain is the human body's primary upper organ. Stroke is a medical disorder in which the blood arteries in the brain are ruptured, causing damage to the brain. When the supply of blood and other nutrients to the brain is interrupted, symptoms might develop. Stroke is considered as medical urgent situation and can cause long-term neurological damage, complications and often death. The World Health Organization (WHO) claims that stroke is the leading cause of death and disability worldwide. Early detection of the numerous stroke warning symptoms can lessen the stroke's severity. The main objective of this study is to forecast the possibility of a brain stroke occurring at an early stage using deep learning and machine learning techniques. To gauge the effectiveness of the algorithm, a reliable dataset for stroke prediction was taken from the Kaggle website. Several classification models, including Extreme Gradient Boosting (XGBoost), Ada Boost, Light Gradient Boosting Machine, Random Forest, Decision Tree, Logistic Regression, K Neighbors, SVM - Linear Kernel, Naive Bayes, and deep neural networks (3-layer and 4-layer ANN) were successfully used in this study for classification tasks. The Random Forest classifier has 99% classification accuracy, which was the highest (among the machine learning classifiers). The three-layer deep neural network (4-Layer ANN) has produced a higher accuracy of 92.39% than the three-layer ANN method utilizing the selected features as input. They findings showed that machine learning techniques outperformed deep neural networks.

Emon MU et al., 2020 proposed the study intitle with Performance analysis of machine learning approaches in stroke prediction. The proposes is an early prediction of stroke diseases by using different machine learning approaches with the occurrence of hypertension, body mass index level, heart disease, average glucose level, smoking status, previous stroke and age. Using these high features attributes, ten different classifiers have been trained, they are Logistics Regression, Stochastic Gradient Descent, Decision Tree Classifier, AdaBoost Classifier, Gaussian Classifier, Quadratic Discriminant Analysis, Multi- layer Perceptron Classifier, KNeighbors Classifier, Gradient Boosting Classifier, and XGBoost Classifier for predicting the stroke. Afterwards, results of the base classifiers are aggregated by using the weighted voting approach to reach highest accuracy. Moreover, the propsoed study has achieved an accuracy of 97%, where the weighted voting classifier performs better than the base classifiers. This model gives the best accuracy for the stroke prediction. The area under curve value of weighted voting classifier is also high. False positive rate and false negative rate of weighted classifier is lowest compared with others. As a result, weighted voting is almost the perfect classifier for predicting the stroke that can be used by physicians and patients to prescribe and early detect a potential stroke.

Sailasya G et al., 2021 proposed the study on Analyzing the performance of stroke prediction using ML classification algorithms**.** A Stroke is a health condition that causes damage by tearing the blood vessels in the brain. It can also occur when there is a halt in the blood flow and other nutrients to the brain. According to the World Health Organization (WHO), stroke is the leading cause of death and disability globally. Most of the work has been carried out on the prediction of heart stroke but very few works show the risk of a brain stroke. With this thought, various machine learning models are built to predict the possibility of stroke in the brain. This paper has taken various physiological factors and used machine learning algorithms like Logistic Regression, Decision Tree Classification, Random Forest Classification, K-Nearest Neighbors, Support Vector Machine and Naive Bayes Classification to train five different models for accurate prediction. The algorithm that best performed this task is Naïve Bayes that gave an accuracy of approximately 82%.

Bandi V et al., 2020 investigated the study on prediction of brain stroke severity using machine learning. Among different types of strokes, ischemic and hemorrhagic majorly damages the central nervous system. According to the World Health Organization (WHO), globally 3% of the population are affected by subarachnoid hemorrhage, 10% with intracerebral hemorrhage, and the majority of 87% with ischemic stroke. In this research work, Machine Learning techniques are applied in identifying, classifying, and predicting the stroke from medical information. The existing research is limited in predicting risk factors pertained to various types of strokes. To address this limitation a Stroke Prediction (SPN) algorithm is proposed by using the improvised random forest in analyzing the levels of risks obtained within the strokes. They concluded that the Stroke Predictor (SPR) model using machine learning techniques improved the prediction accuracy to 96.97% when compared with the existing models.

Mahesh KA et al., 2020 proposed the study on predicting of stroke using machine learning Stroke is a blood clot or bleeds in the brain, which can make permanent damage that has an effect on mobility, cognition, sight or communication. Stroke is considered as medical urgent situation and can cause long-term neurological damage, complications and often death. The majority of strokes are classified as ischemic embolic and Hemorrhagic. An ischemic embolic stroke happens when a blood clot forms away from the patient brain usually in the patient heart and travels through the patient bloodstream to lodge in narrower brain arteries. Hemorrhagic stroke is considered another type of brain stroke as it happens when an artery in the brain leaks blood or ruptures. Stroke is the second leading cause of death worldwide and one of the most life- threatening diseases for persons above 65 years. It injures the brain like “heart attack” which injures the heart. Once a stroke disease occurs, it is not only cost huge medical care and permanent disability but can eventually lead to death. Every 4 minutes someone dies of stroke, but up to 80% of stroke can be prevented if we can identify or predict the occurrence of stroke in its early stage.

Objective

To develop and evaluate machine learning models that predict the risk of stroke by leveraging clinical and lifestyle data. This study aims to enhance the early detection of stroke, allowing for timely intervention and prevention strategies.

- Gather a comprehensive dataset comprising clinical and lifestyle factors associated with stroke risk. Perform data cleaning, normalization, and handling of missing values to prepare the data for analysis.
- Conduct EDA to identify key patterns, trends, and correlations within the dataset. Visualize the distribution of risk factors and their relationship with stroke incidence
- Develop multiple machine learning models (e.g., Logistic Regression, Decision Tree, Random Forest, Support Vector Machine) for stroke prediction.
- Evaluate the performance of each model using metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC).

Chapter 2

Exploratory Data Analysis

This chapter is divided into six parts, these are data description, data preprocessing, exploratory data analysis (EDA), machine learning algorithms, Evaluation matrices and working flowchart. The implantation procedure is explained in detail along with necessary figures in this section.

2.1 Data Description

This dataset has been collected from the medical clinic of Bangladesh to estimate whether a patient is likely to suffer from a stroke. It is the document of 599 people's information including 11 attributes (Performance Analysis of Machine Learning Approaches in Stroke Prediction Minhaz Uddin Emon, Maria Sultana Keya, Tamara Islam Meghla, Md. Mahfujur Rahman, MShamim Al Mamun, and M Shamim Kaiser).

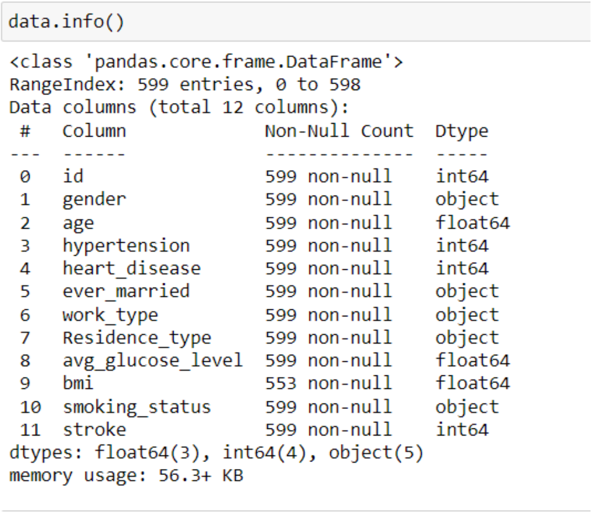
- **id:** This attribute refers to the id of the patient.
- **age:** This attribute means a person's age. It's numerical data.
- **gender:** This attribute means a person's gender. It's categorical data involving male, female
- **hypertension:** This attribute means whether a person is hypertensive or not. 0 is for the patient who does not have hypertension, and 1 is for the patient who has hypertension. It is numerical data.
- **work_type:** This attribute represents the person's work scenario. It is categorical data involving children, Gov-jobs, never worked, Private or Self-employed.

- **residence_type:** This attribute represents the person's living scenario. It is categorical data involving Rural or Urban.
- **heart_disease:** This attribute means whether a person has heart disease or not. 0 is for the patient who does not have any heart disease, and 1 is for the patient who has a heart disease. It is numerical data.
- **avg_glucose_level**: This attribute represents the average glucose level in blood of a patient level in blood. It is numerical data.
- **bmi:** This attribute means body mass index of a person. It's numerical data.
- **ever_married:** this attribute represents a person's marital status. It is categorical data involving No or Yes.
- **Smoking_status:** This attribute represents a person's smoking status. It is categorical data involving formerly smoked, never smoked, smokes, or Unknown.
- **stroke:** This attribute means a person previously had a stroke or not. It is numerical data. 1 means that the patient had a stroke and 0 is the opposite.

Among all these attributes, stroke is the decision class, and the rest of the attribute is the response class.

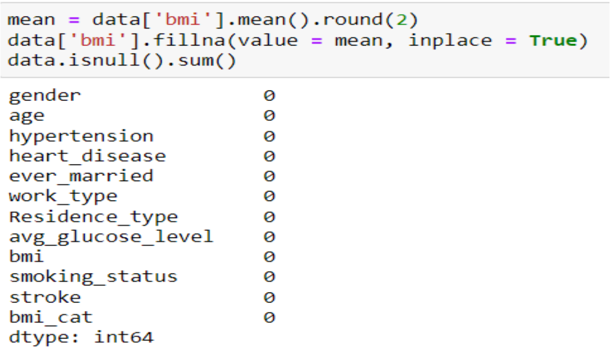
2.2 Data Preprocessing

In data preprocessing there are several steps which must go through, the 1st one is Null checking. In programming, there can be events in which a variable may not be assigned a value. Such events are frequently represented by the unique value null. The dataset's null or missing values can be checked using the Pandas function is null. Only the rows with null values are presented since the null values are assigned to true values. In this study, some null values are found in the dataset. These null values are filled by the mean value using the fillna function.



```
data.info()
```

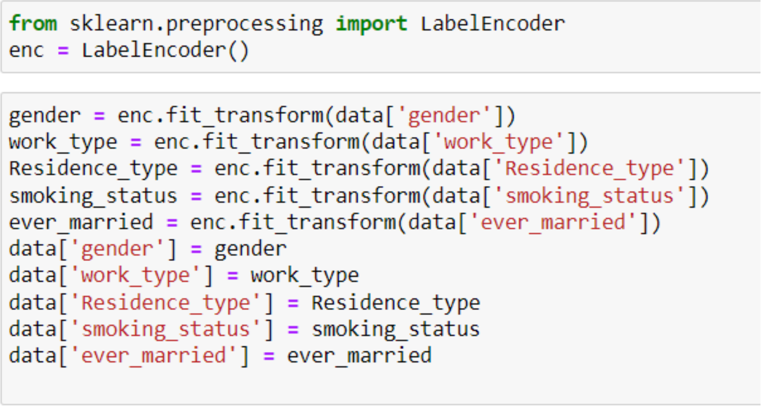
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 599 entries, 0 to 598
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 599 non-null    int64  
 1   gender             599 non-null    object 
 2   age                599 non-null    float64
 3   hypertension       599 non-null    int64  
 4   heart_disease      599 non-null    int64  
 5   ever_married       599 non-null    object 
 6   work_type          599 non-null    object 
 7   Residence_type     599 non-null    object 
 8   avg_glucose_level  599 non-null    float64
 9   bmi                553 non-null    float64
 10  smoking_status     599 non-null    object 
 11  stroke             599 non-null    int64  
dtypes: float64(3), int64(4), object(5)
memory usage: 56.3+ KB
```



```
mean = data['bmi'].mean().round(2)
data['bmi'].fillna(value = mean, inplace = True)
data.isnull().sum()
```

```
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                  0
smoking_status       0
stroke               0
bmi_cat              0
dtype: int64
```

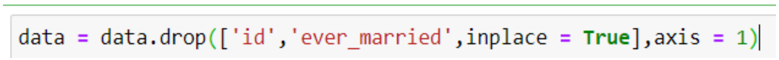
Another step is Converting categorical data into numerical data. The data set's categorical variables should be transformed into numerical values. Several machine learning algorithms can support categorical values naturally, one is LabelEncoder form scikit-learn package. (Geeks for Geeks, 2024).



```
from sklearn.preprocessing import LabelEncoder
enc = LabelEncoder()
```

```
gender = enc.fit_transform(data['gender'])
work_type = enc.fit_transform(data['work_type'])
Residence_type = enc.fit_transform(data['Residence_type'])
smoking_status = enc.fit_transform(data['smoking_status'])
ever_married = enc.fit_transform(data['ever_married'])
data['gender'] = gender
data['work_type'] = work_type
data['Residence_type'] = Residence_type
data['smoking_status'] = smoking_status
data['ever_married'] = ever_married
```

Further we are going to drop id and ever_married column form the dataset as they are not important for our analysis.

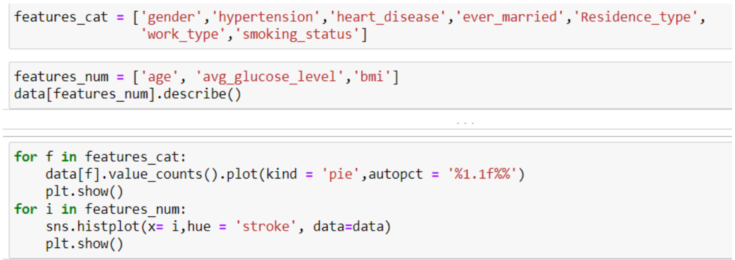


```
data = data.drop(['id','ever_married',inplace = True],axis = 1)
```

2.3 Exploratory Data Analysis (EDA)

EDA is performed to understand the relationships between variables and the target variable

3.3.1 Univariate Analysis: Analyze individual features using pie chart, histograms, etc.



```
features_cat = ['gender','hypertension','heart_disease','ever_married','Residence_type',
                'work_type','smoking_status']
```

```
features_num = ['age', 'avg_glucose_level','bmi']
data[features_num].describe()
```

```
for f in features_cat:
    data[f].value_counts().plot(kind = 'pie',autopct = '%1.1f%%')
    plt.show()
for i in features_num:
    sns.histplot(x= i,hue = 'stroke', data=data)
    plt.show()
```

2.3.2 Bivariate Analysis: Explore associations between two variables, especially with the target variable, using correlation matrices, scatter plots, and cross-tabulations.

Crosstabs: To describe a single categorical variable, we use frequency tables. To describe the relationship between two categorical variables, we use a special type of table called a cross-tabulation (or "crosstab" for short). In a cross-tabulation, the categories of one variable determine the rows of the table, and the categories of the other variable determine the columns. The cells of the table contain the number of times that a particular combination of categories occurred. The "edges" (or "margins") of the table typically contain the total number of observations for that category.

Assumptions

1. Your two variables should be measured at an ordinal or nominal level (i.e., categorical data).

2. Your two variables should consist of two or more categorical, independent groups.

Hypothesis: The null hypothesis (Ho) and alternative hypothesis (H_1) of the Chi-Square Test of Independence can be expressed as-

Ho: "[Variable 1] is not significantly associated with [Variable 2]"

H_1: "[Variable 1] is significantly associated with [Variable 2]"

Test Statistic: The test statistic for the Chi-Square Test of Independence is denoted χ^2 and is computed as:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{((o_{ij} - e_{ij})^2}{e_{ij}}$$

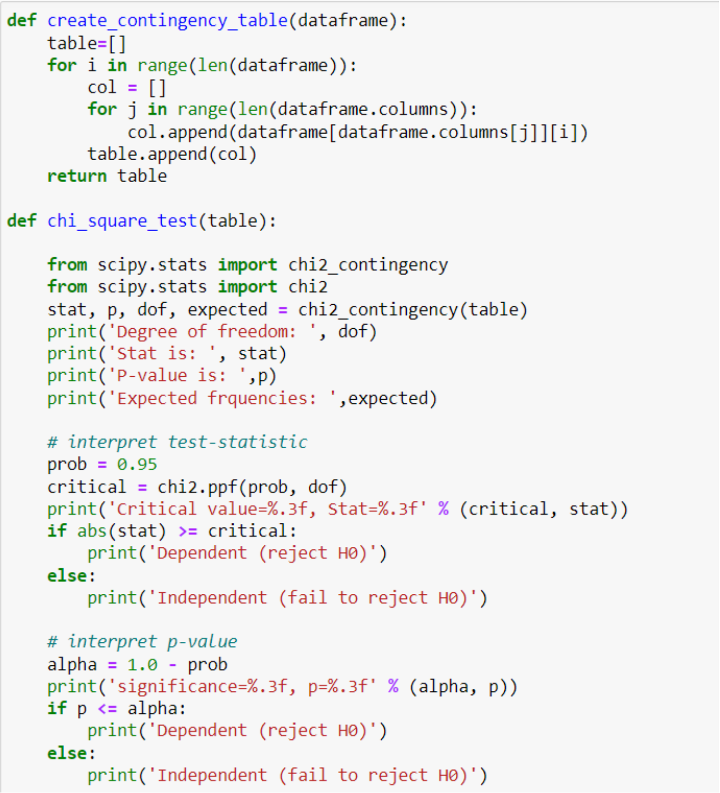
were,

o_{ij} is the observed cell count in the ith row and jth column of the table

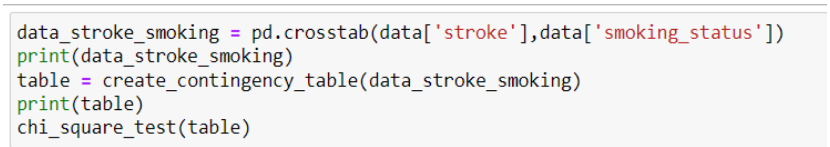
e_{ij} is the expected cell count in the i^{th} row and j^{th} column of the table, computed as:

$$e_{ij} = \frac{e_{ij}\ is\ i^{th}\ row\ total\ and\ j^{th} column\ total}{grand\ total}$$

The quantity ($o_{ij} – e_{ij}$) is sometimes referred to as the residual of cell (i, j), denoted r_{ij}. The calculated χ^2 value is then compared to the critical value from the χ^2 distribution table with degrees of freedom df = (r - 1)(c - 1) and chosen confidence level. If the calculated χ^2 value > critical χ^2 value, then we reject the null hypothesis.



```
def create_contingency_table(dataframe):
    table=[]
    for i in range(len(dataframe)):
        col = []
        for j in range(len(dataframe.columns)):
            col.append(dataframe[dataframe.columns[j]][i])
        table.append(col)
    return table

def chi_square_test(table):

    from scipy.stats import chi2_contingency
    from scipy.stats import chi2
    stat, p, dof, expected = chi2_contingency(table)
    print('Degree of freedom: ', dof)
    print('Stat is: ', stat)
    print('P-value is: ',p)
    print('Expected frquencies: ',expected)

    # interpret test-statistic
    prob = 0.95
    critical = chi2.ppf(prob, dof)
    print('Critical value=%.3f, Stat=%.3f' % (critical, stat))
    if abs(stat) >= critical:
        print('Dependent (reject H0)')
    else:
        print('Independent (fail to reject H0)')

    # interpret p-value
    alpha = 1.0 - prob
    print('significance=%.3f, p=%.3f' % (alpha, p))
    if p <= alpha:
        print('Dependent (reject H0)')
    else:
        print('Independent (fail to reject H0)')
```



```
data_stroke_smoking = pd.crosstab(data['stroke'],data['smoking_status'])
print(data_stroke_smoking)
table = create_contingency_table(data_stroke_smoking)
print(table)
chi_square_test(table)
```

2.4 Working Flowchart

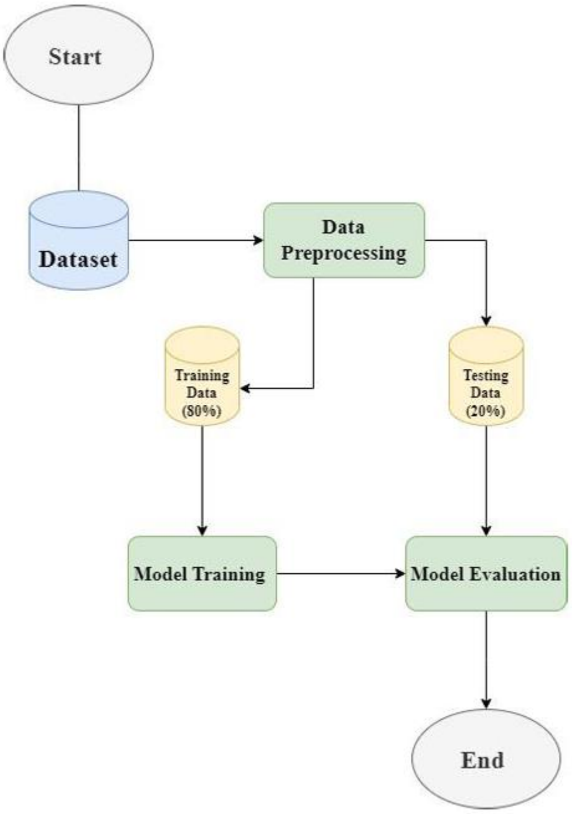


Fig 2.1 Working Flowchart of this study

This coding part of this work was done in Python. Thus, all the necessary python and Scikit-learn libraries are utilized. After collecting the data, the preprocess is to make it suitable for further use. After that, the dataset is split into two parts, i.e. training data and testing. The training dataset is trained using the three classifiers. The testing data is evaluated using the same classifiers to measure its performance. After analyzing the result, the best algorithm is identified.

Exploratory data analysis

Descriptive and Graphs

Brief description of data through frequency tables and pie charts

- Frequency tables and pie charts for Nominal Variable
- Histogram for continues variable

Gender

gender	Female	Male	Total
Frequency	355	244	599
percentage	59.27%	40.73%	100.00%

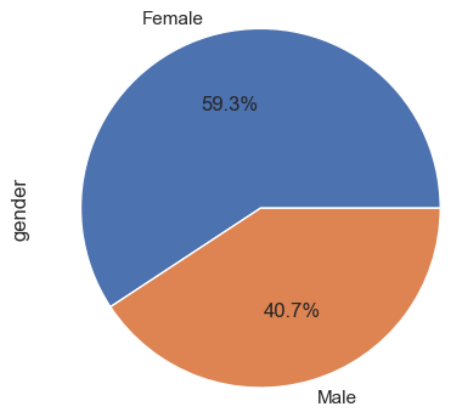


Fig-2.2

Proportion of females is More than males

Hypertension

Hypertension	Yes (1)	No (0)	Total
Frequency	98	501	599
Percentage	16.36%	83.64%	100.00%

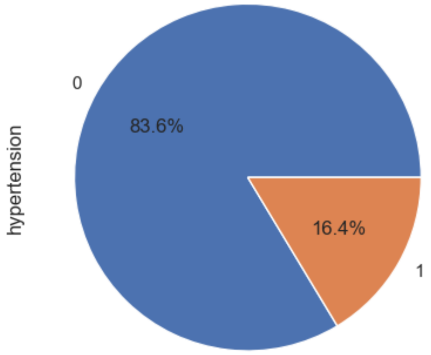


Fig-2.3

Proportion of patients not having hypertension is more than patients having hypertension

Heart-disease

Heart_disease	Yes (1)	No (0)	Total
Frequency	62	537	599
Percentage	10.35%	89.65%	100.00%

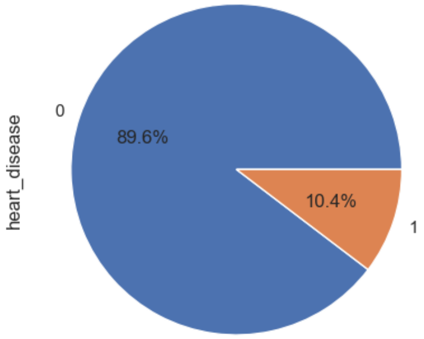


Fig-2.4

Proportion of patients not having heart disease is more than patients having heart disease

Residence-Type

Residence_type	Rural	Urban	Total
Frequency	293	306	599
Percentage	48.91%	51.09%	100.00%

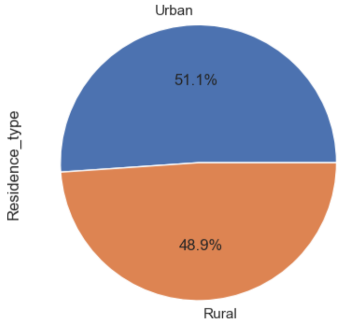


Fig-2.5

Proportion of patients living in rural area or patients lives in urban area are approximately equal

Work-type

Work_type	Gov_job	Never_worked	Private	Self-employed	Children	Total
Frequency	69	1	356	125	48	599
Percentage	11.52%	0.17%	59.43%	20.87%	8.01%	100.00%

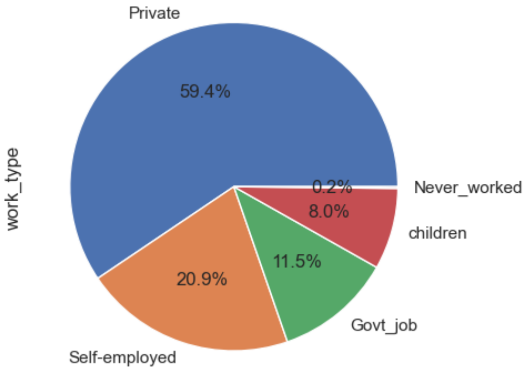


Fig-2.6

Proportion of patients working in private sector are much higher followed by self-employed, children, govt_job, and patients who never worked are less

Smoking Status

Smoking_status	Unknown	Formerly_smoked	Never_smoked	Smokes	Total
Frequency	157	128	215	99	599
Percentage	26.21%	21.37%	35.89%	16.53%	100.00%

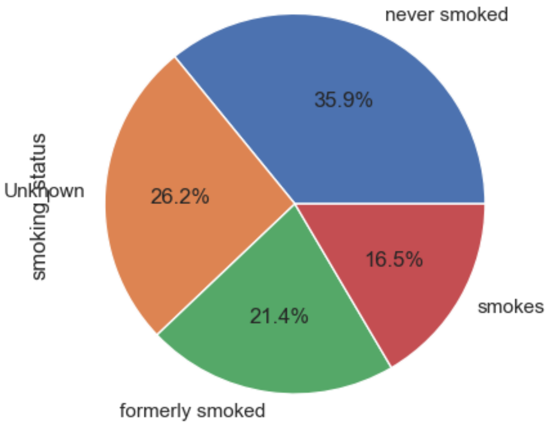
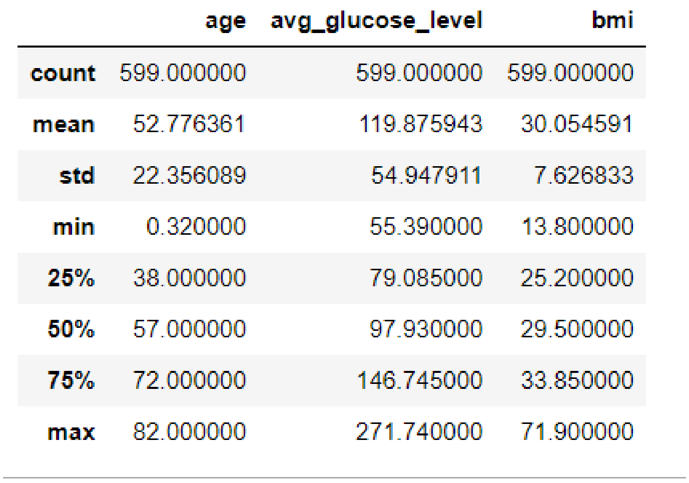


Fig-2.7

Proportion of patients who never smoked are more followed by unknown, formerly smoked, and patients who smokes are less than all

Descriptive Statistics



	age	avg_glucose_level	bmi
count	599.000000	599.000000	599.000000
mean	52.776361	119.875943	30.054591
std	22.356089	54.947911	7.626833
min	0.320000	55.390000	13.800000
25%	38.000000	79.085000	25.200000
50%	57.000000	97.930000	29.500000
75%	72.000000	146.745000	33.850000
max	82.000000	271.740000	71.900000

The table provides summary statistics for three variables across 599 observations: age, average glucose level, and BMI. The mean age is approximately 52.78 years, with a standard deviation of 22.36 years. The average glucose level has a mean of 119.88 mg/dL and a standard deviation of 54.95 mg/dL, while the BMI has a mean of 30.05 with a standard deviation of 7.63. The data ranges from ages 0.32 to 82 years, glucose levels from 55.39 to 271.74 mg/dL, and BMI from 13.8 to 71.9.

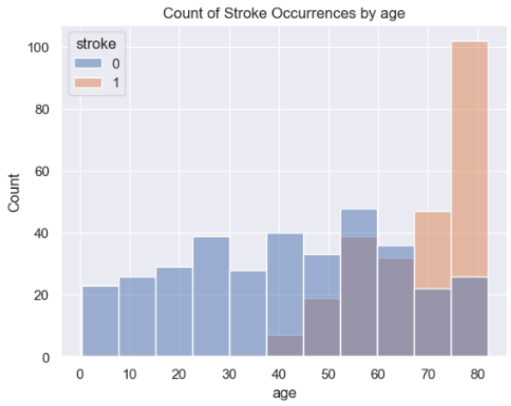


Fig-2.8

The bar graph shows the count of stroke occurrences by age group. Younger age groups (0-50 years) have fewer stroke occurrences, with a relatively low incidence in those under 40. The number of stroke cases increases significantly in older age groups, particularly from age 60 onwards, peaking sharply at age 80. This suggests a strong correlation between advancing age and the likelihood of experiencing a stroke.

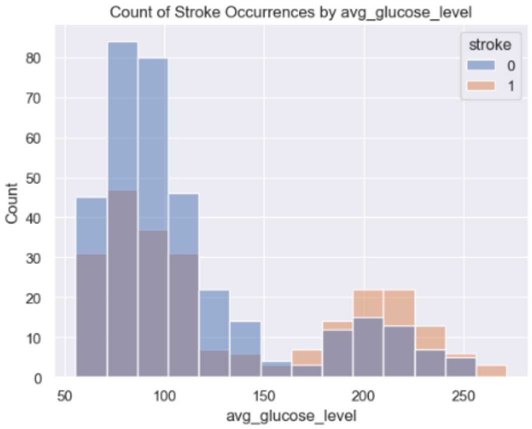


Fig-2.9

The bar chart illustrates the count of stroke occurrences categorized by average glucose level. Individuals with lower average glucose levels (around 70-100) have a higher count of non-stroke cases. As average glucose levels increase, the count of stroke occurrences rises, particularly between the 150-200 range. This trend suggests a potential link between higher average glucose levels and an increased risk of stroke.

4.2 Bivariate analysis

Gender and Stroke

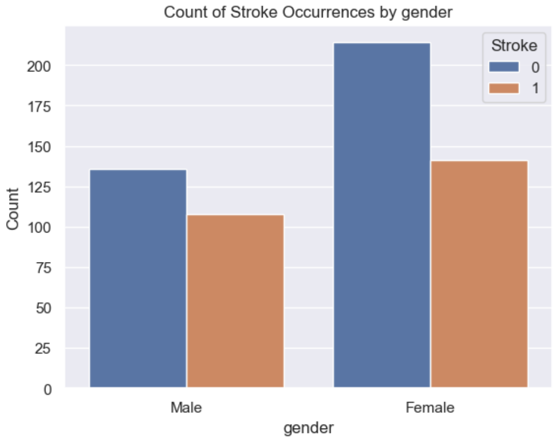


Fig-2.10

Stroke	Gender				Total
	Female		Male		
	n	%	n	%	
NO	214	60.28%	136	55.74%	350
YES	141	39.72%	108	44.26%	249
Total	355	100.00%	244	100.00%	599
Chi-square test	χ^2=1.04; p=0.306				

H_0: There is no association between stroke and gender

H_1: There is association between stroke and gender

The chi-square test results show that there is no significant association between gender and stroke occurrence (χ2 = 1.04, p = 0.306). The percentage of females and males who did not experience a stroke are 60.28% and 55.74%, respectively, while those who experienced a stroke are 39.72% for females and 44.26% for males. Since the p-value is greater than 0.05, we fail to reject the null hypothesis, indicating that gender does not significantly influence stroke occurrence in this dataset.

Hypertension and Stroke

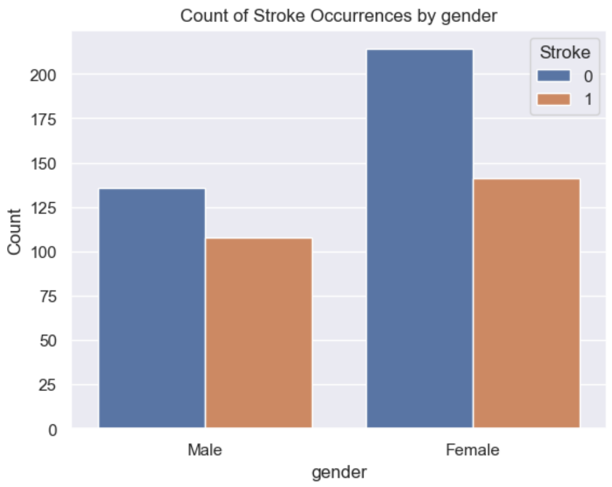


Fig-2.11

Stroke	Hypertension				Total
	No	%	Yes	%	
No	318	63.47%	32	86.75	350
Yes	183	36.53%	66	13.25	249
Total	501	100	98	100	599
Chi-square test	χ^2=30.798 p=0.00				

H0: There is no association between stroke and hypertension

H1: There is association between stroke and hypertension

The chi-square test results indicate a significant association between hypertension and stroke occurrence (χ2 = 30.798, p = 0.00). Among individuals without hypertension, 63.47% did not experience a stroke, while 36.53% did. In contrast, among those with hypertension, 86.75% did not experience a stroke, while only 13.25% did. Since the p-value is less than 0.05, we reject the null hypothesis, indicating that hypertension significantly influences stroke occurrence in this dataset.

Heart_disease and Stroke

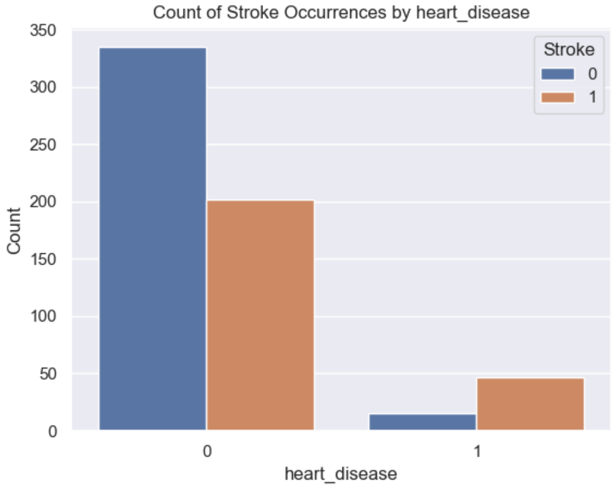


Fig-.12

stroke	heart_disease				
	No	**%**	**Yes**	**%**	**Total**
0	335	62.38%	15	24.19%	350
1	202	37.62%	47	75.81%	249
Total	537	100%	62	100%	599
Chi-square test	$\chi^{2=}$ 31.822 p=0.00				

H$_0$: There is no association between stroke and heart_disease

H$_1$: There is association between stroke and heart_disease

The chi-square test results indicate a significant association between heart disease and stroke occurrence (χ2 = 31.822, p = 0.00). Among individuals without heart disease, 62.38% did not experience a stroke, while 37.62% did. Conversely, among those with heart disease, 22.95% did not experience a stroke, while a notable 77.05% did. Since the p-value is less than 0.05, we reject the null hypothesis, indicating that heart disease significantly influences stroke occurrence in this dataset.

Residence_type and Stroke

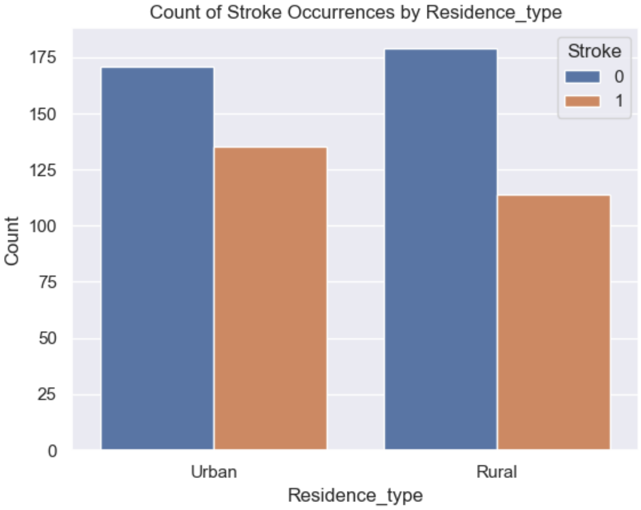


Fig-2.13

Stroke	**residence_type**				Total
	Rural	**%**	**Urban**	~~%~~	
0	179	61.09%	171	55.88%	350
1	114	38.91%	135	44.12%	249
Total	293	100%	306	100%	599
Chi-square test	χ2 = 1.465 p=0.226				

H$_0$: There is no association between stroke and residence_type

H$_1$: There is association between stroke and residence_type

The chi-square test results indicate no significant association between residence type and stroke occurrence (χ2 = 1.465, p = 0.226). Among individuals residing in rural areas, 61.09% did not experience a stroke, while 38.91% did. For those living in urban areas, 55.88% did not experience a stroke, and 44.12% did. Since the p-value is greater than 0.05, we fail to reject the null hypothesis, suggesting that residence type does not significantly influence stroke occurrence in this dataset.

Work_type and Stroke

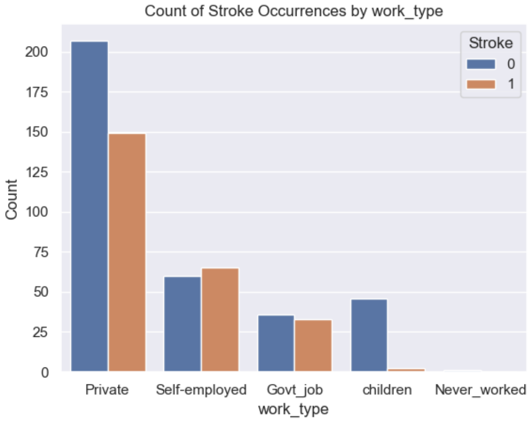


Fig-2.14

stroke	work_type										Total
	Govt_job	%	Never_worked	%	Private	%	Self-employed	%	children	%	
0	36	52.17%	1	100%	207	58.15%	60	48%	46	95.83%	350
1	33	47.83%	0	0%	149	41.85%	65	52%	2	4.17%	249
Total	69	100%	1	100%	356	100%	125	100%	48	100%	599
chi-square test	χ2 = 49.164 p-value = 0.00										

H_0: There is no association between stroke and work_type

H_1: There is association between stroke and work_type

The chi-square test results show a significant association between work type and stroke occurrence (χ2 = 49.164, p-value = 0.00). In the dataset, those with government jobs have a nearly equal distribution of stroke occurrence (52.17% no stroke, 47.83% stroke). Those who never worked had no stroke occurrences. Among private sector employees, 58.15% did not have strokes, while 41.85% did. For self-employed individuals, 48% did not experience strokes, while 52% did. Children had the highest percentage of no stroke occurrence at 95.83%, with only 4.17% experiencing strokes. Since the p-value is less than 0.05, we reject the null hypothesis, indicating that work type significantly influences stroke occurrence in this dataset.

Smoking_status and Stroke

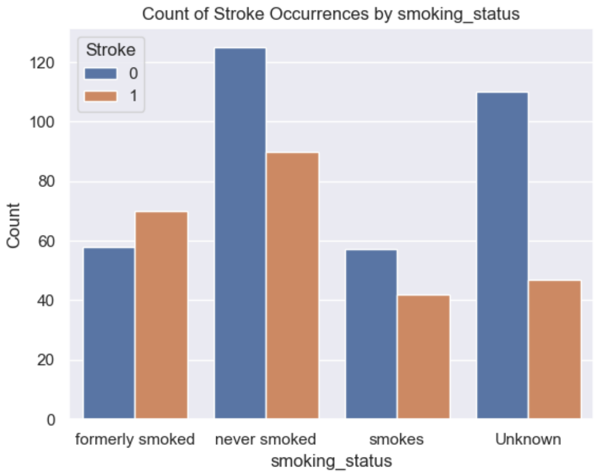


Fig-2.15

stroke	smoking_status								Total
	Unknown	%	Formerly_smoked	%	Never_smoked	%	smokes	%	
0	110	70.06%	58	45.31%	125	58.14%	57	57.58%	350
1	47	29.94%	70	54.69%	90	41.86%	42	42.42%	249
Total	157	100%	128	100%	215	100%	99	100%	599
Chi-square test	$\chi 2$ = 17.853 p-value = 0.00								

H_0: There is no association between stroke and work_type

H_1: There is association between stroke and work_type

The chi-square test results indicate a significant association between smoking status and stroke occurrence ($\chi 2$ = 17.853, p-value = 0.00). For individuals with an unknown smoking status, 70.06% did not have a stroke, while 29.94% did. Among former smokers, 45.31% did not have a stroke, whereas 54.69% did, indicating a higher stroke occurrence compared to non-smokers. Those who never smoked showed 58.14% without stroke and 41.86% with stroke. For current smokers, 57.58% did not experience strokes, and 42.42% did. The significant p-value (less than 0.05) leads us to reject the null hypothesis, suggesting that smoking status is significantly associated with stroke occurrence in this dataset.

BMI_cat and stroke

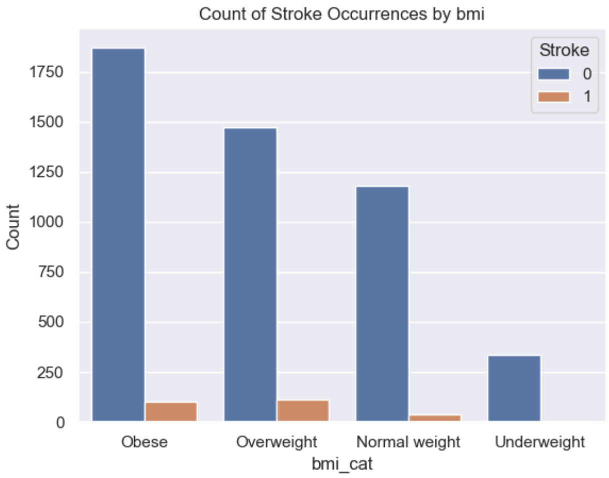


Fig-2.16

stroke	bmi_cat								Total
	Normal weight	%	Obese	%	Overweight	%	Underweight	%	
0	78	69.03 %	151	51.71 %	97	57.40 %	24	96 %	350
1	35	30.97 %	141	48.29 %	72	42.60 %	1	4%	249
Total	113	100 %	292	100 %	169	100 %	25	100 %	599
Chi-square test	χ2 = 25.251 p-value = 0.00								

H$_0$: There is no association between stroke and work_type

H$_1$: There is association between stroke and work_type

The chi-square test results indicate a significant association between BMI category and stroke occurrence (χ2 = 25.251, p-value = 0.00). Among individuals with normal weight, 69.03% did not have a stroke, while 30.97% did. For those categorized as obese, 51.71% did not have a stroke, whereas 48.29% did, suggesting a higher stroke occurrence in this group. Individuals classified as overweight showed 57.40% without stroke and 42.60% with stroke. Among the underweight group, 96% did not experience strokes, and only 4% did. The significant p-value (less than 0.05) leads us to reject the null hypothesis, indicating that BMI category is significantly associated with stroke occurrence in this dataset.

Chapter-3

Prediction of Brain Stroke using Logistic Regression

Machine Learning

Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that can learn from data and generalize to unseen data, and thus perform tasks without explicit instructions. Recently, artificial neural networks have been able to surpass many previous approaches in performance. ML finds application in many fields, including natural language processing, computer vision, speech recognition, email filtering, agriculture, and medicine. When applied to business problems, it is known under the name predictive analytics. Although not all machine learning is statistically based, computational statistics is an important source of the field's methods. The mathematical foundations of ML are provided by mathematical optimization (mathematical programming) methods. Data mining is a related (parallel) field of study, focusing on exploratory data analysis (EDA) through unsupervised learning.

Machine learning methods:

Machine learning models fall into three primary categories.

Supervised machine learning

Supervised learning, also known as supervised machine learning, is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately. As input data is fed into the model, the model adjusts its weights until it has been fitted appropriately. This occurs as part of the cross-validation process to ensure that the model avoids overfitting or underfitting. Supervised learning helps organizations solve a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning include neural networks, naïve bayes, linear regression, logistic regression, random forest, and support vector machine (SVM).

Unsupervised machine learning

Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets (subsets called clusters). These algorithms discover hidden patterns or data groupings without the need for human intervention. This method's ability to discover similarities and differences in information make it ideal for exploratory data analysis, cross-selling strategies, customer segmentation, and image and pattern recognition. It's also used to reduce the number of features in a model through the process of dimensionality reduction. Principal component analysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include neural networks, k-means clustering, and probabilistic clustering methods.

Semi-supervised learning

Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labeled data set to guide classification and feature extraction from a larger, unlabeled data set. Semi-supervised learning can solve the problem of not having enough labeled data for a supervised learning algorithm. It also helps if it's too costly to label enough data.

Reinforcement machine learning

Reinforcement machine learning is a machine learning model that is similar to supervised learning, but the algorithm isn’t trained using sample data. This model learns as it goes by using trial and error. A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem.

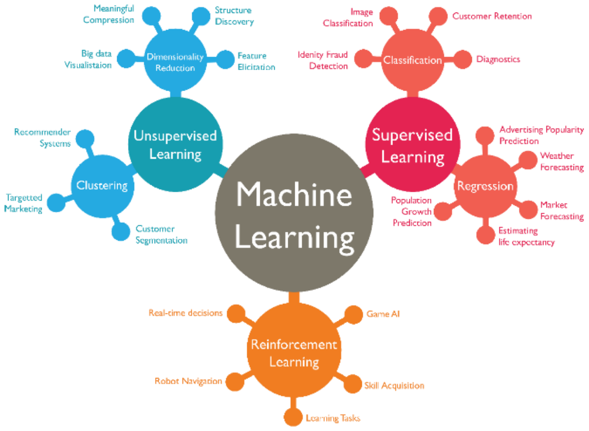
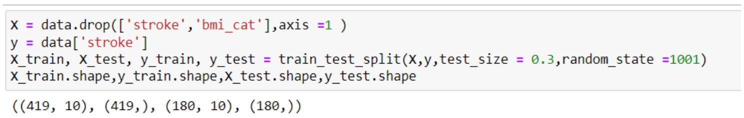


Fig 3.1 Types of Machine Learning

Train-Test Split

The Train-Test Split technique was adopted to evaluate the performance of a model. It divides some percentage of the data for training the model and the rest for testing the model so that it can compare my machine-learning model to the actual machine-learning. The model selection module is provided in the sci-kit-learn library in which the function train_test_split resideds. In our model, we have split the training set into 70% and the test set into 30% of the dataset. The last step is Feature scaling. A technique for normalizing the number of independent variables or features in data is called feature scaling. It is typically carried out during the data preprocessing step and is sometimes referred to as data normalization in the context of data processing. Each feature's value in the data is standardized so that it has a zero mean and unit variance. The fundamental approach to computation is to identify the distribution mean and standard deviation for each feature, then use the formula below to generate the new data point.



```
X = data.drop(['stroke','bmi_cat'],axis =1 )
y = data['stroke']
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.3,random_state =1001)
X_train.shape,y_train.shape,X_test.shape,y_test.shape
```

```
((419, 10), (419,), (180, 10), (180,))
```

Evaluation Metrics

Metrics for model evaluation play a role in evaluating the performance and accuracy of a model. These metrics help in evaluating the model's performance and in comparing it to other models or algorithms. In this work, the model is evaluated using the Confusion Matrix and its helpful measurements, such as Accuracy, Precision, Recall and F1-score. In Confusion Matrix, the evaluation of a machine learning model's performance on a set of test data is summarized by a confusion matrix. It is frequently used for measuring how well categorization models work. These models try to predict a categorical label for each input event. The matrix shows how many true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) the model generated using the test data. True denotes a precise prediction of the values, while False denotes an incorrect prediction. In True Positive, the number of times our real positive values match our positive predictions. In False Positive, the number of times a model mispredicts positive values as negatives. Although it was expected a negative value, the outcome is positive. In True Negative, the number of times our real negative values match our negative predictions. In False Negative, the number of times a model predicts positive values for negative values in error. It is positive, contrary to what I had predicted. In accuracy, the percentage of values that were correctly categorized is determined using accuracy. It is the result of dividing the sum of all true values by all values.

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}}$$

where, TP, TN, FP, and FN stand for True Positive, True Negative, False Positive and False Negative respectively.

In precision, the model's accuracy in classifying positive values is determined by precision. It is the ratio of the true positives to all the predicted positive values.

$$\boldsymbol{Precision = \frac{TP}{TP + FP}}$$

where, TP and FP stand for True Positive and False Positive respectively.

In Recall, it is a measurement of the percentage of true positive cases (or actual positive cases) that were correctly predicted as positive. Another name for the recall is sensitivity. This suggests that there will be a further percentage of actual positive cases that are mistakenly forecasted as negative (and are hence sometimes referred to as the false negative). A false negative rate can also be used to demonstrate this.

$$\boldsymbol{Recall = \frac{TP}{TP + FN}}$$

where, TP and FN stand for True Positive and False Negative respectively. A higher recall number would indicate a higher true positive and a lower false negative. Lower recall would translate to lower true positive and larger false negative values. Models with high sensitivity are preferred for the healthcare and finance industries.

In F1-Score, Recall and precision are effectively combined to form this. When both precision and sensitivity must be considered, it is helpful.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

ROC Curve

An ROC curve, or receiver operating characteristic curve, is like a graph that shows how well a classification model performs. It helps us see how the model makes decisions at different levels of certainty. The curve has two lines: one for how often the model correctly identifies positive cases (true positives) and another for how often it mistakenly identifies negative cases as positive (false positives). By looking at this graph, we can understand how good the model is and choose the threshold that gives us the right balance between correct and incorrect predictions.

The **Receiver Operator Characteristic (ROC)** curve is an evaluation metric for binary classification problems. It is a probability curve that plots the **TPR** against **FPR** at various threshold values and essentially **separates the 'signal' from the 'noise.'** In other words, it shows the performance of a classification model at all classification thresholds. The **Area Under the Curve (AUC)** is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the model's performance at distinguishing between the positive and negative classes.

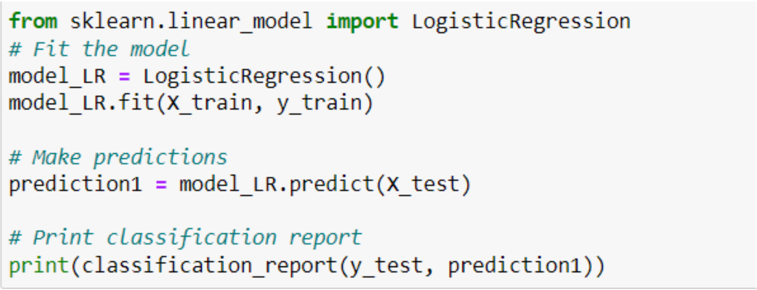
Logistic Regression

This type of statistical model (also known as *logit model*) is often used for classification and predictive analytics. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:

$$Logit(\pi) \; = \frac{1}{(1 + e^{-\pi})}$$

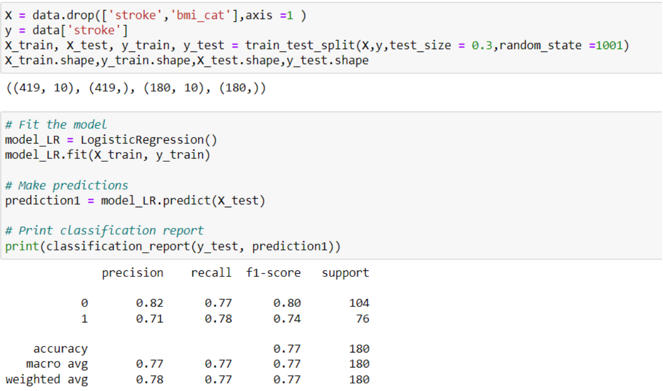
$$\ln\left(\frac{\pi}{1 - \pi}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

In this logistic regression equation, logit(π) is the dependent or response variable and x is the independent variable. The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE). This method tests different values of beta through multiple iterations to optimize for the best fit of log odds. All of these iterations produce the log likelihood function, and logistic regression seeks to maximize this function to find the best parameter estimate. Once the optimal coefficient (or coefficients if there is more than one independent variable) is found, the conditional probabilities for each observation can be calculated, logged, and summed together to yield a predicted probability. For binary classification, a probability less than .5 will predict 0 while a probability greater than 0 will predict 1. After the model has been computed, it's best practice to evaluate the how well the model predicts the dependent variable, which is called goodness of fit. The Hosmer–Lemeshow test is a popular method to assess model fit. By importing the LogisticRegression class from the sklearn.liner_model library, the model is fit to the training set.



```
from sklearn.linear_model import LogisticRegression
# Fit the model
model_LR = LogisticRegression()
model_LR.fit(X_train, y_train)

# Make predictions
prediction1 = model_LR.predict(X_test)

# Print classification report
print(classification_report(y_test, prediction1))
```

Fig-3.2



```
X = data.drop(['stroke','bmi_cat'],axis =1 )
y = data['stroke']
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.3,random_state =1001)
X_train.shape,y_train.shape,X_test.shape,y_test.shape
```

```
((419, 10), (419,), (180, 10), (180,))
```

```
# Fit the model
model_LR = LogisticRegression()
model_LR.fit(X_train, y_train)

# Make predictions
prediction1 = model_LR.predict(X_test)

# Print classification report
print(classification_report(y_test, prediction1))
```

```
              precision    recall  f1-score   support

           0       0.82      0.77      0.80       104
           1       0.71      0.78      0.74        76

    accuracy                           0.77       180
   macro avg       0.77      0.77      0.77       180
weighted avg       0.78      0.77      0.77       180
```

Fig-3.3

The performance metrics for the logistic regression model indicate a balanced performance between the two classes. For the negative class (0), the model achieved a precision of 0.82, a recall of 0.77, and an F1-score of 0.80, indicating a high correctness of positive predictions and a substantial proportion of actual negatives being correctly identified. For the positive class (1), the precision is 0.71, recall is 0.78, and the F1-score is 0.74, demonstrating a moderate level of correctness for positive predictions and a relatively high proportion of actual positives being correctly identified. The overall accuracy of the model is 0.77, with a macro average and weighted average F1-score both at 0.77, indicating a consistent performance across both classes.

ROC curve for Logistic Regression

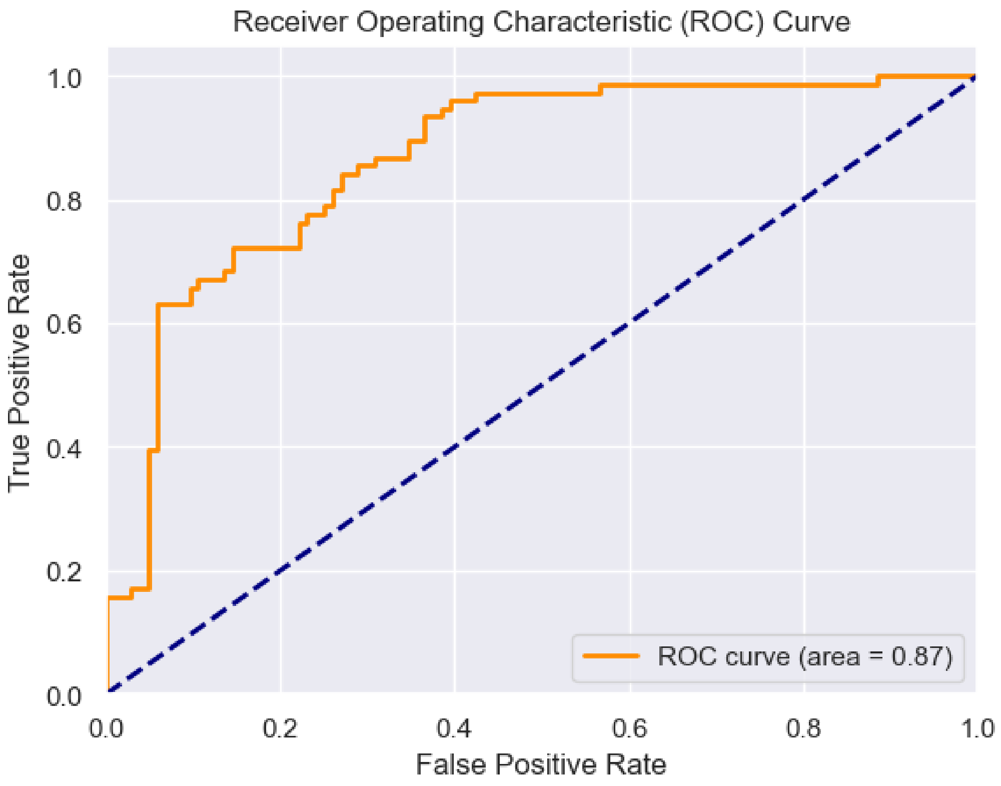


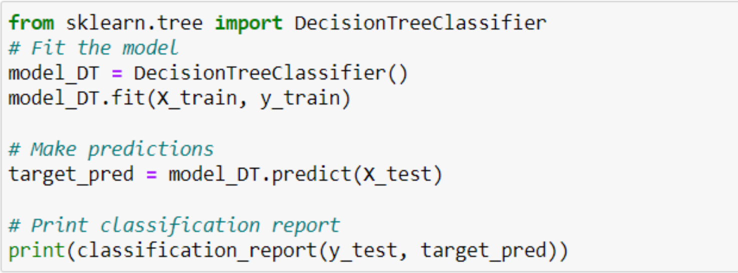
Fig-3.4

The ROC curve shows that the logistic regression model has a good discriminative ability with an AUC of 0.87. This means that the model is effective at distinguishing between the positive and negative classes, as a higher AUC value (closer to 1) indicates better performance.

Chapter-4

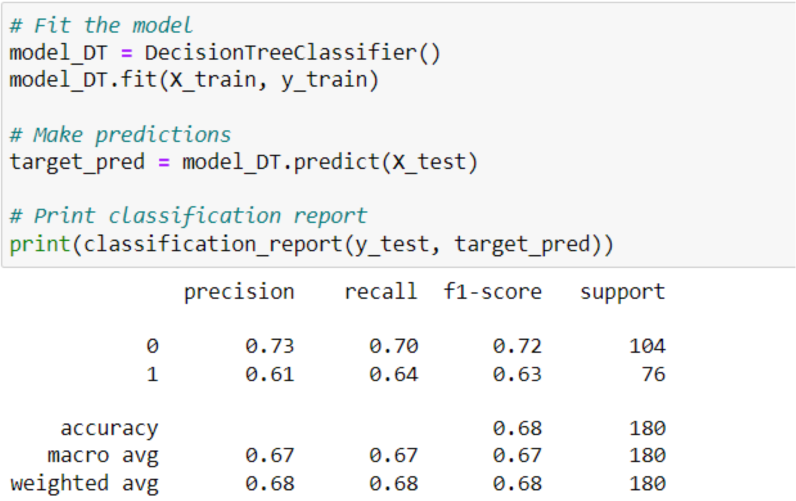
Prediction of Brain Stroke using Decision Tree Model

Decision Tree, a decision support method that resembles a tree is called a decision tree. It has three parts decision nodes, leaf nodes, and a root node. A training dataset is divided into branches by a decision tree algorithm, which then further splits the branches into other branches. This pattern keeps on until a leaf node is reached. Further separation of the leaf node is not possible. The attributes utilized to forecast the outcome are represented by the nodes in the decision tree. Links to the leaves are provided by the decision nodes. The most important thing to keep in mind while developing a machine learning model is to select the optimal method for the dataset and task at hand. The two main benefits of adopting a decision tree are that it frequently simulates human decision-making processes, making it simple to understand and that the decision tree's justification is clear because it has a tree-like structure. By importing the DecisionTreeClassifier class from the sklearn.tree library, the model is fit to the training set. The code is below



```
from sklearn.tree import DecisionTreeClassifier
# Fit the model
model_DT = DecisionTreeClassifier()
model_DT.fit(X_train, y_train)

# Make predictions
target_pred = model_DT.predict(X_test)

# Print classification report
print(classification_report(y_test, target_pred))
```

Fig-4.1



```
# Fit the model
model_DT = DecisionTreeClassifier()
model_DT.fit(X_train, y_train)

# Make predictions
target_pred = model_DT.predict(X_test)

# Print classification report
print(classification_report(y_test, target_pred))
```

```
              precision    recall  f1-score   support

           0       0.73      0.70      0.72       104
           1       0.61      0.64      0.63        76

    accuracy                           0.68       180
   macro avg       0.67      0.67      0.67       180
weighted avg       0.68      0.68      0.68       180
```

Fig-4.2

The performance metrics for the decision tree model indicate moderate effectiveness. For the negative class (0), the model achieved a precision of 0.73, a recall of 0.70, and an F1-score of 0.72, showing a relatively high accuracy in identifying actual negatives. For the positive class (1), the precision is 0.61, recall is 0.64, and the F1-score is 0.63, indicating moderate performance in identifying actual positives. The overall accuracy of the model is 0.68, with both the macro and weighted averages at 0.67-0.68, suggesting that the model's performance is fairly consistent across both classes but slightly better at predicting no-stroke cases.

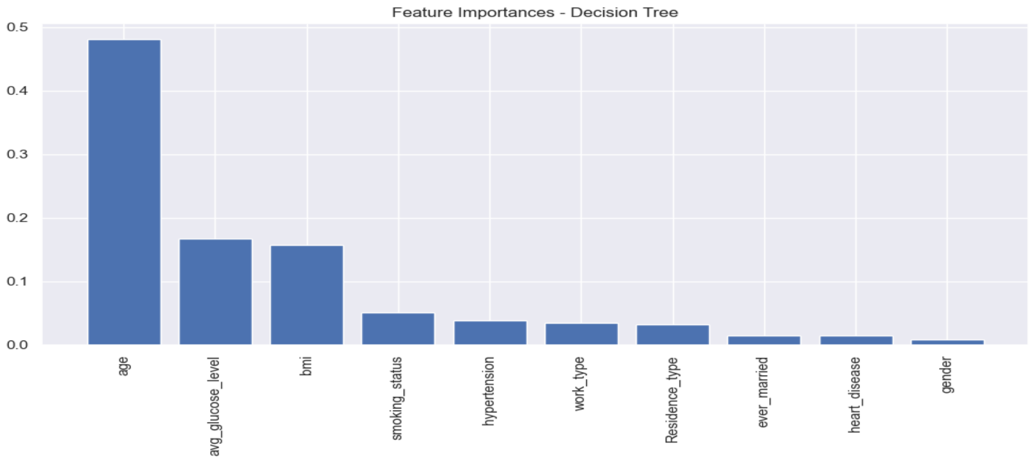


Fig-4.3

The feature importances for the decision tree model indicate that **age** (48.10%) is the most critical factor in predicting the target variable, followed by **average glucose level** (16.74%) and **BMI** (15.67%). **smoking status** (5.10%) also have moderate importance. Other features such as **gender**, **residence type**, **ever married**, **hypertension**, **work type** and **heart disease** have lower contributions, each accounting for less than 5% of the model's decisions. This highlights that demographic factors, especially age, and health metrics like glucose level and BMI, are the key drivers in the model's predictive performance.

ROC curve of Decision Tree

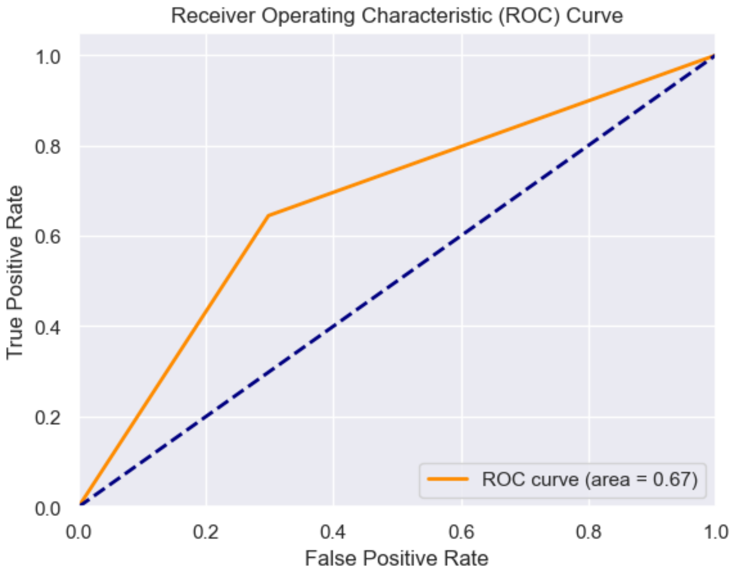


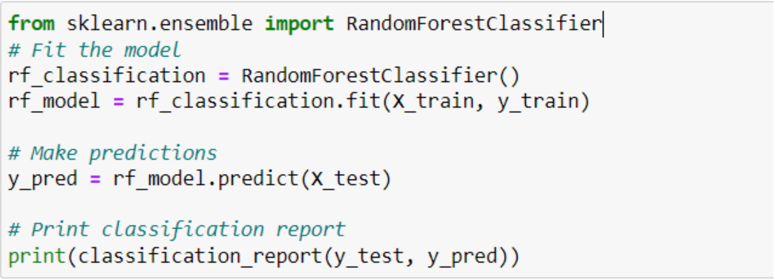
Fig-4.4

The ROC curve shown indicates the performance of a decision tree model. The curve plots the True Positive Rate (sensitivity) against the False Positive Rate, and the area under the curve (AUC) is 0.67. An AUC of 0.67 suggests that the model has moderate discriminative ability, performing better than random chance (AUC = 0.5) but not achieving high accuracy (AUC close to 1).

Chapter-5

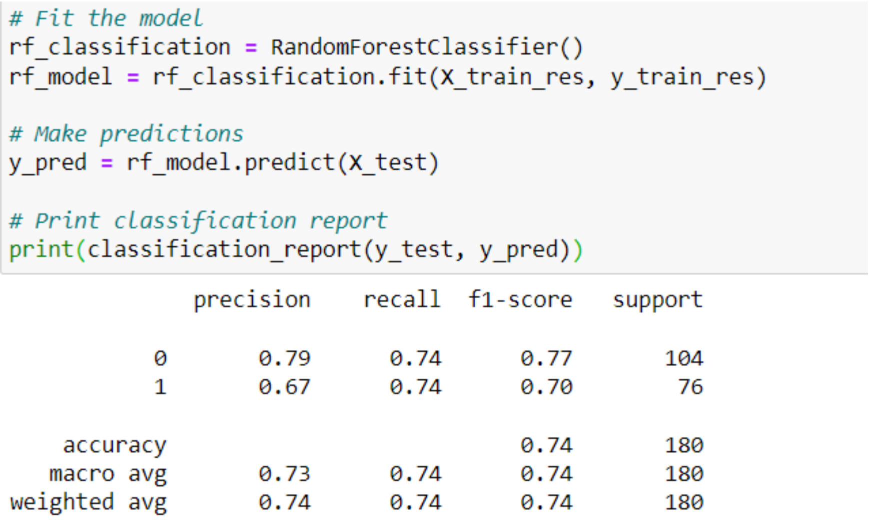
Prediction of Brain Stroke using Random Forest Classifier

In a random forest algorithm, there are many different decision trees. The random forest algorithm creates a forest that is trained via bagging or bootstrap aggregation. 15 Based on the predictions made by the decision trees, this algorithm determines the outcome. It makes predictions by averaging out the results from different trees. The accuracy of the result grows as the number of trees increases. The decision tree algorithm's shortcomings are eliminated with a random forest. It improves precision and lowers dataset overfitting. Many different decision trees are used in a random forest system. Each decision tree has a root node, leaf node, and decision node. The result generated by a particular decision tree is represented by the leaf node of each tree. The majority voting method is used to choose the result. The final output of the system in this scenario is the output that most of the decision trees have selected. Random Forest systems are used by medical practitioners to diagnose patients. Patients are diagnosed based on an evaluation of their past medical experience. To determine the appropriate dosage for the patients, prior medical records are examined. By importing the Random Forest Classifier class from sklearn. ensemble library, the Random Forest algorithm is fit to the training data. The code is below



```python
from sklearn.ensemble import RandomForestClassifier
# Fit the model
rf_classification = RandomForestClassifier()
rf_model = rf_classification.fit(X_train, y_train)

# Make predictions
y_pred = rf_model.predict(X_test)

# Print classification report
print(classification_report(y_test, y_pred))
```

Fig-5.1



```
# Fit the model
rf_classification = RandomForestClassifier()
rf_model = rf_classification.fit(X_train_res, y_train_res)

# Make predictions
y_pred = rf_model.predict(X_test)

# Print classification report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.79      0.74      0.77       104
           1       0.67      0.74      0.70        76

    accuracy                           0.74       180
   macro avg       0.73      0.74      0.74       180
weighted avg       0.74      0.74      0.74       180
```

Fig-5.2

The classification report for the random forest model indicates an overall accuracy of 74%. The model performs better on class 0, with a precision of 0.79 and an F1-score of 0.77, compared to class 1, which has a precision of 0.67 and an F1-score of 0.70. The macro and weighted averages for precision, recall, and F1-score are all around 0.74, suggesting a balanced performance across both classes but with room for improvement, particularly in distinguishing class 1 instances.

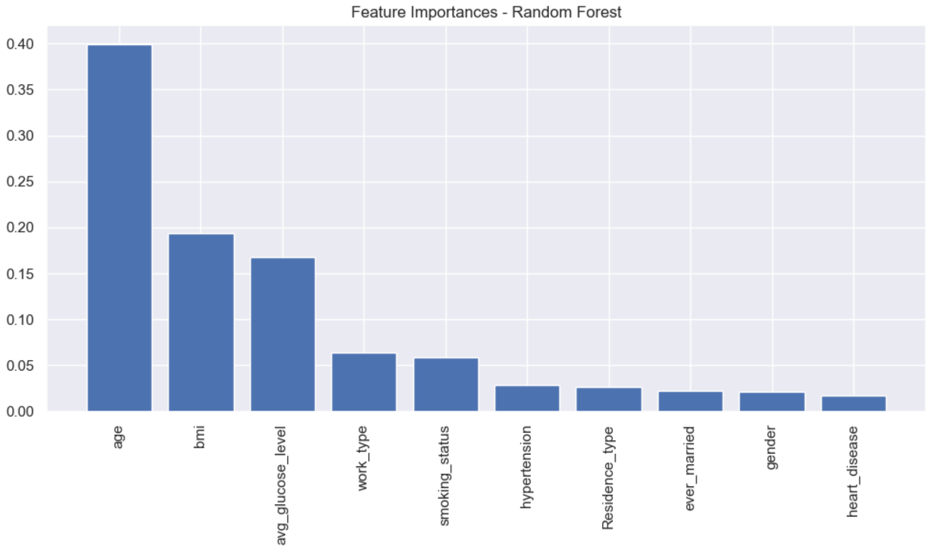


Fig-5.3

The feature importances for the random forest model indicate that **age** (39.95%) is the most influential factor in predicting the target variable, followed by BMI (19.36%) and **average glucose level** (16.82%). **Work type** (6.42%) and **smoking status** (5.88%) also play moderate roles. Other features like **gender**, **residence type**, **ever married**, **hypertension**, and **heart disease** contribute less significantly, each accounting for less than 5% of the model's decisions. This distribution suggests that demographic and health-related factors, particularly age, glucose level, and BMI, are crucial in the model's predictive performance.

<u>ROC Curve for Random Forest</u>

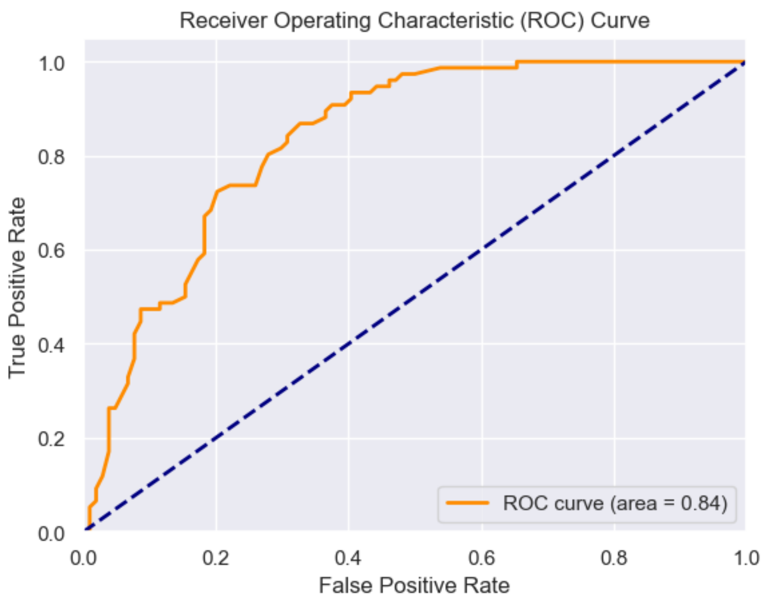


Fig-5.4

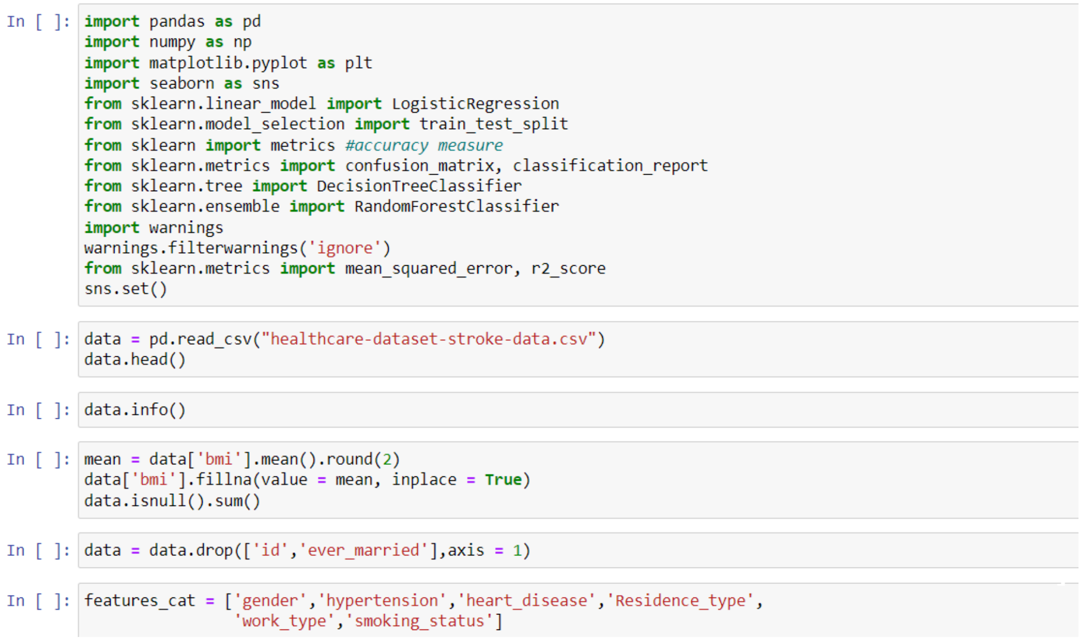
The ROC curve for the random forest model shows a high level of performance with an area under the curve (AUC) of 0.84. This indicates that the model has a strong ability to distinguish between the positive and negative classes, performing significantly better than random chance (AUC = 0.5). The high AUC suggests the model is effective at predicting the true positives while maintaining a low false positive rate.

Conclusion

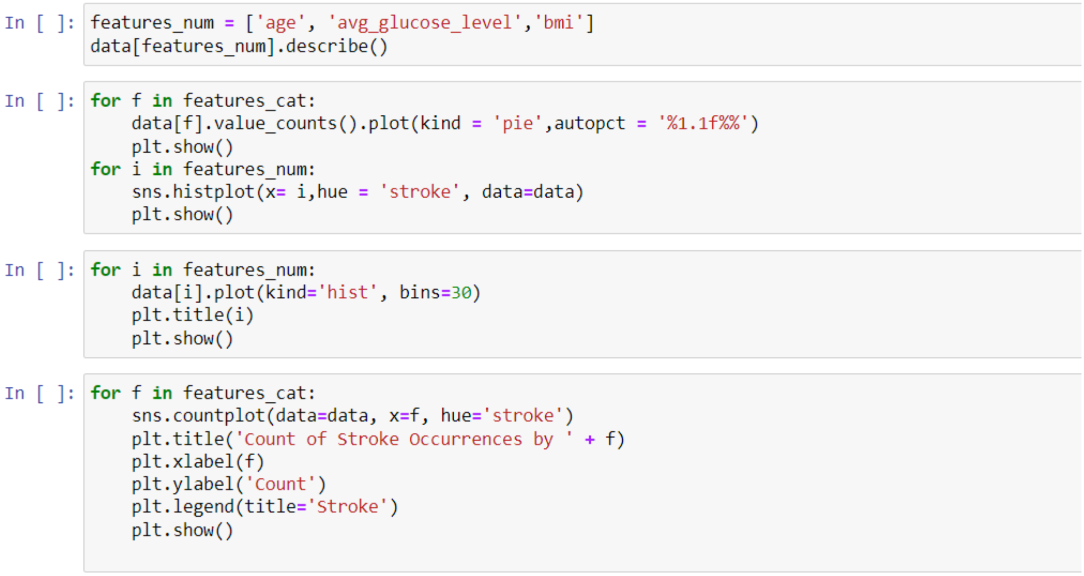
Metric	Logistic Regression	Decision Tree	Random Forest
Accuracy	0.77	0.68	0.74
Precision (Class 0)	0.82	0.73	0.79
Precision (Class 1)	0.71	0.61	0.67
Recall (Class 0)	0.77	0.7	0.74
Recall (Class 1)	0.78	0.64	0.74
F1-Score (Class 0)	0.8	0.72	0.77
F1-Score (Class 1)	0.74	0.63	0.7
ROC AUC	0.87	0.67	0.84

In comparing the three models for predicting brain stroke, logistic regression stands out as the most effective, achieving the highest accuracy of 77% and an impressive ROC AUC of 0.87, which indicates a strong ability to distinguish between positive and negative cases. The random forest model also demonstrates robust performance with a 74% accuracy and a ROC AUC of 0.84, making it a viable alternative. In contrast, the decision tree model underperforms relative to the others, with a lower accuracy of 68% and a ROC AUC of 0.67, suggesting it is less reliable for this prediction task.

Appendix



```
In [ ]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.linear_model import LogisticRegression
        from sklearn.model_selection import train_test_split
        from sklearn import metrics #accuracy measure
        from sklearn.metrics import confusion_matrix, classification_report
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier
        import warnings
        warnings.filterwarnings('ignore')
        from sklearn.metrics import mean_squared_error, r2_score
        sns.set()

In [ ]: data = pd.read_csv("healthcare-dataset-stroke-data.csv")
        data.head()

In [ ]: data.info()

In [ ]: mean = data['bmi'].mean().round(2)
        data['bmi'].fillna(value = mean, inplace = True)
        data.isnull().sum()

In [ ]: data = data.drop(['id','ever_married'],axis = 1)

In [ ]: features_cat = ['gender','hypertension','heart_disease','Residence_type',
                        'work_type','smoking_status']
```



```
In [ ]: features_num = ['age', 'avg_glucose_level','bmi']
        data[features_num].describe()

In [ ]: for f in features_cat:
            data[f].value_counts().plot(kind = 'pie',autopct = '%1.1f%%')
            plt.show()
        for i in features_num:
            sns.histplot(x= i,hue = 'stroke', data=data)
            plt.show()

In [ ]: for i in features_num:
            data[i].plot(kind='hist', bins=30)
            plt.title(i)
            plt.show()

In [ ]: for f in features_cat:
            sns.countplot(data=data, x=f, hue='stroke')
            plt.title('Count of Stroke Occurrences by ' + f)
            plt.xlabel(f)
            plt.ylabel('Count')
            plt.legend(title='Stroke')
            plt.show()
```

```
In [ ]: def create_contingency_table(dataframe):
            table=[]
            for i in range(len(dataframe)):
                col = []
                for j in range(len(dataframe.columns)):
                    col.append(dataframe[dataframe.columns[j]][i])
                table.append(col)
            return table

        def chi_square_test(table):

            from scipy.stats import chi2_contingency
            from scipy.stats import chi2
            stat, p, dof, expected = chi2_contingency(table)
            print('Degree of freedom: ', dof)
            print('Stat is: ', stat)
            print('P-value is: ',p)
            print('Expected frquencies: ',expected)

            # interpret test-statistic
            prob = 0.95
            critical = chi2.ppf(prob, dof)
            print('Critical value=%.3f, Stat=%.3f' % (critical, stat))
            if abs(stat) >= critical:
                print('Dependent (reject H0)')
            else:
                print('Independent (fail to reject H0)')

            # interpret p-value
            alpha = 1.0 - prob
            print('significance=%.3f, p=%.3f' % (alpha, p))
            if p <= alpha:
                print('Dependent (reject H0)')
            else:
                print('Independent (fail to reject H0)')
```

Stroke and smoking_status

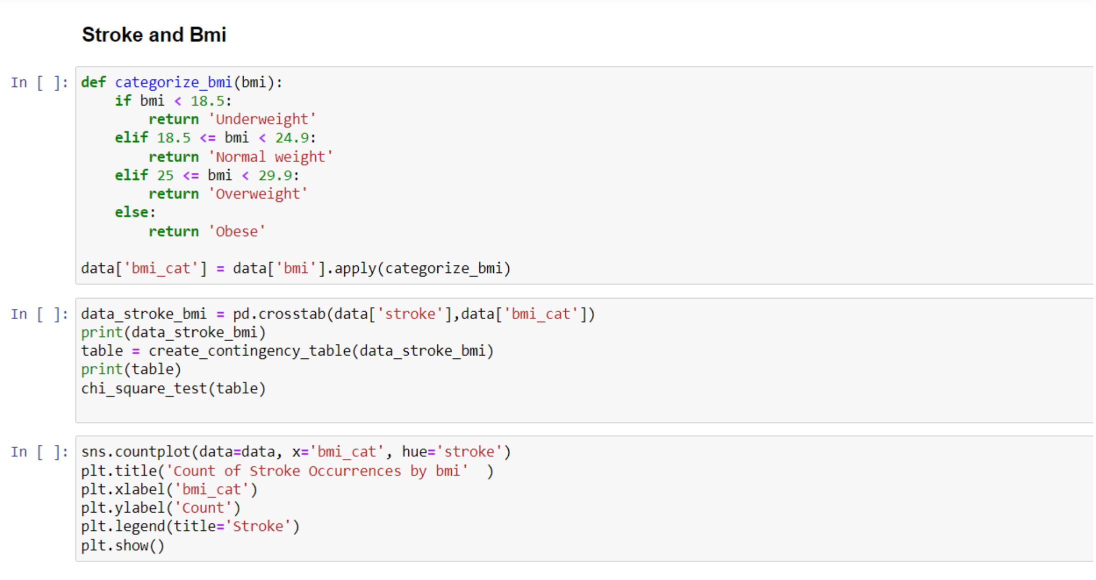
```
In [ ]: data_stroke_smoking = pd.crosstab(data['stroke'],data['smoking_status'])
        print(data_stroke_smoking)
        table = create_contingency_table(data_stroke_smoking)
        print(table)
        chi_square_test(table)
```

Stroke and work_type

```
In [ ]: data_stroke_work = pd.crosstab(data['stroke'],data['work_type'])
        print(data_stroke_work)
        table = create_contingency_table(data_stroke_work)
        print(table)
        chi_square_test(table)
```

Stroke and hypertension

```
In [ ]: data_stroke_hypertension = pd.crosstab(data['stroke'],data['hypertension'])
        print(data_stroke_hypertension)
        table = create_contingency_table(data_stroke_hypertension)
        print(table)
        chi_square_test(table)
```



Stroke and Bmi

In [ ]:

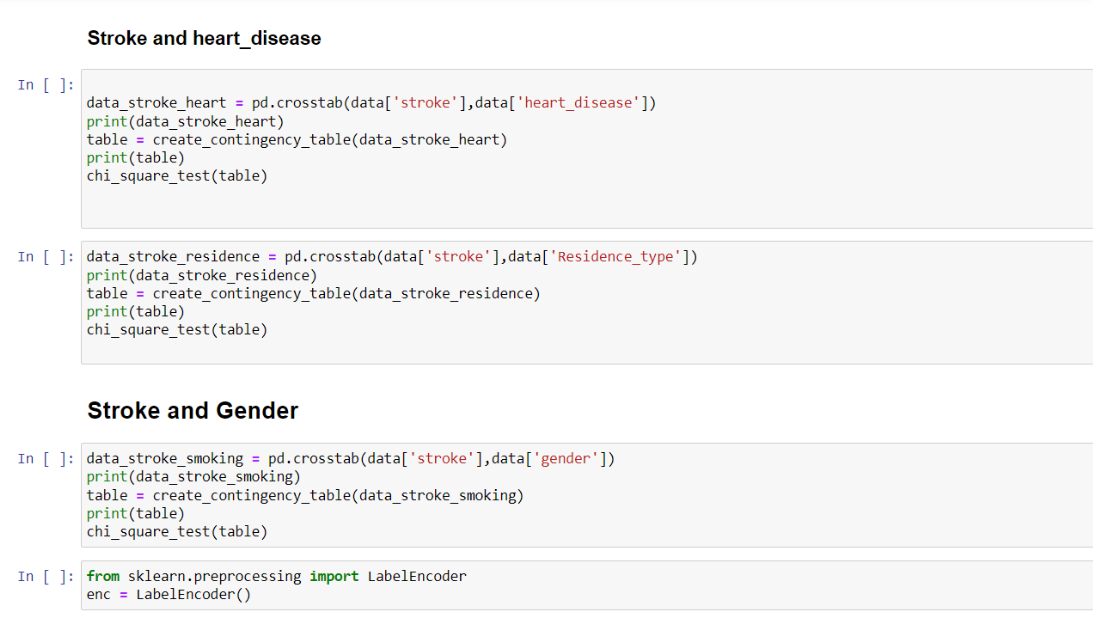
```python
def categorize_bmi(bmi):
    if bmi < 18.5:
        return 'Underweight'
    elif 18.5 <= bmi < 24.9:
        return 'Normal weight'
    elif 25 <= bmi < 29.9:
        return 'Overweight'
    else:
        return 'Obese'

data['bmi_cat'] = data['bmi'].apply(categorize_bmi)
```

In [ ]:

```python
data_stroke_bmi = pd.crosstab(data['stroke'],data['bmi_cat'])
print(data_stroke_bmi)
table = create_contingency_table(data_stroke_bmi)
print(table)
chi_square_test(table)
```

In [ ]:

```python
sns.countplot(data=data, x='bmi_cat', hue='stroke')
plt.title('Count of Stroke Occurrences by bmi'   )
plt.xlabel('bmi_cat')
plt.ylabel('Count')
plt.legend(title='Stroke')
plt.show()
```



Stroke and heart_disease

In [ ]:

```python

data_stroke_heart = pd.crosstab(data['stroke'],data['heart_disease'])
print(data_stroke_heart)
table = create_contingency_table(data_stroke_heart)
print(table)
chi_square_test(table)
```

In [ ]:

```python
data_stroke_residence = pd.crosstab(data['stroke'],data['Residence_type'])
print(data_stroke_residence)
table = create_contingency_table(data_stroke_residence)
print(table)
chi_square_test(table)
```

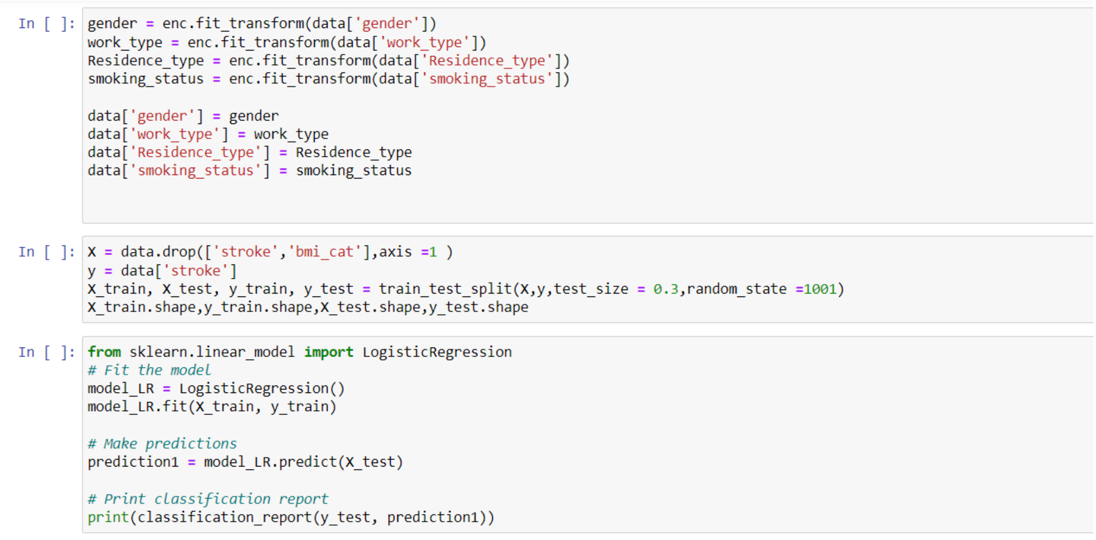
Stroke and Gender

In [ ]:

```python
data_stroke_smoking = pd.crosstab(data['stroke'],data['gender'])
print(data_stroke_smoking)
table = create_contingency_table(data_stroke_smoking)
print(table)
chi_square_test(table)
```

In [ ]:

```python
from sklearn.preprocessing import LabelEncoder
enc = LabelEncoder()
```



In [ ]:

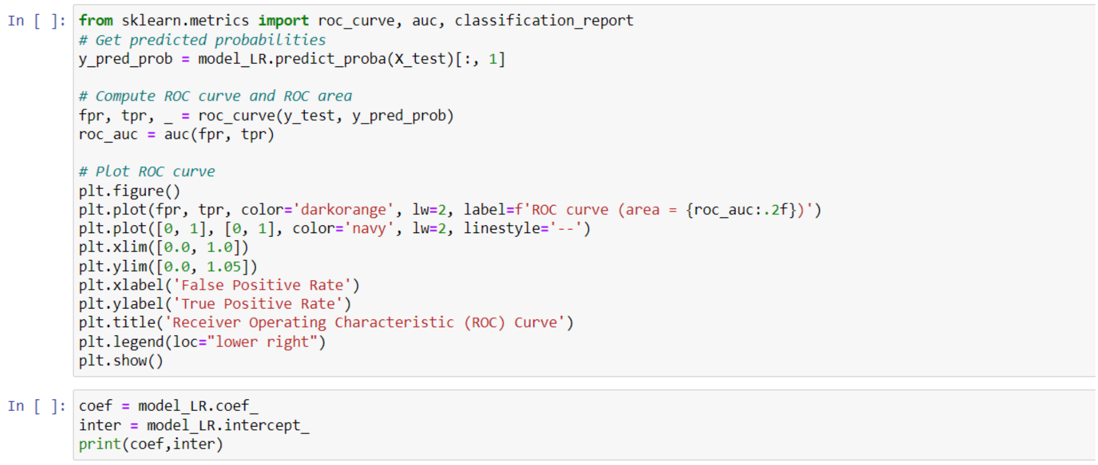
```python
gender = enc.fit_transform(data['gender'])
work_type = enc.fit_transform(data['work_type'])
Residence_type = enc.fit_transform(data['Residence_type'])
smoking_status = enc.fit_transform(data['smoking_status'])

data['gender'] = gender
data['work_type'] = work_type
data['Residence_type'] = Residence_type
data['smoking_status'] = smoking_status
```

In [ ]:

```python
X = data.drop(['stroke','bmi_cat'],axis =1 )
y = data['stroke']
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.3,random_state =1001)
X_train.shape,y_train.shape,X_test.shape,y_test.shape
```

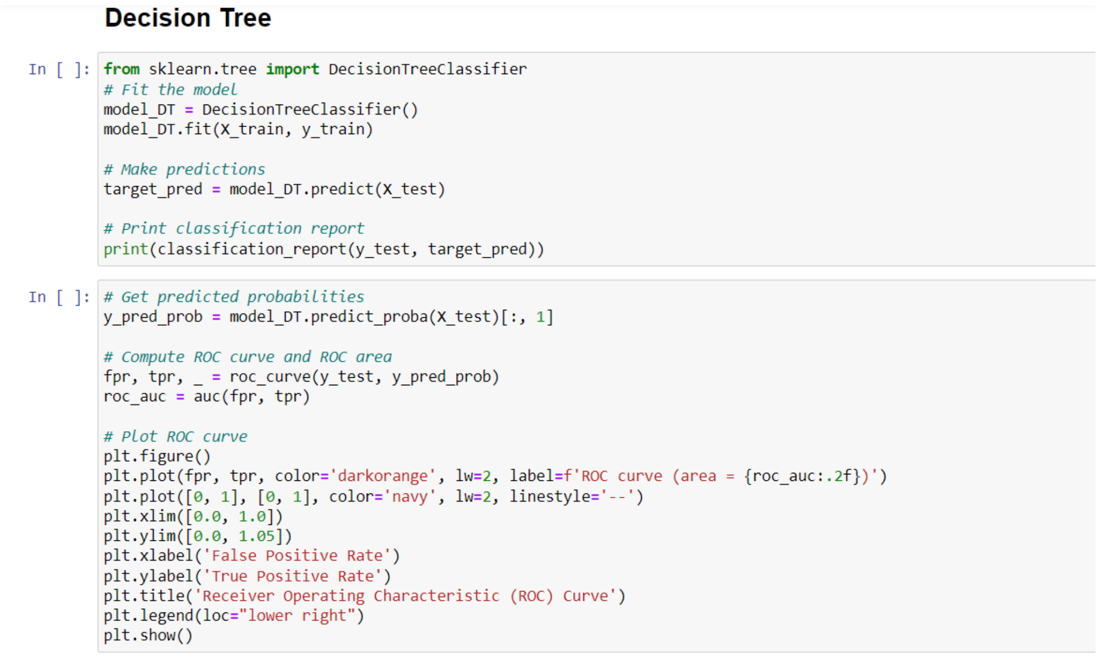
In [ ]:

```python
from sklearn.linear_model import LogisticRegression
# Fit the model
model_LR = LogisticRegression()
model_LR.fit(X_train, y_train)

# Make predictions
prediction1 = model_LR.predict(X_test)

# Print classification report
print(classification_report(y_test, prediction1))
```



```
In [ ]: from sklearn.metrics import roc_curve, auc, classification_report
        # Get predicted probabilities
        y_pred_prob = model_LR.predict_proba(X_test)[:, 1]

        # Compute ROC curve and ROC area
        fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
        roc_auc = auc(fpr, tpr)

        # Plot ROC curve
        plt.figure()
        plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
        plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
        plt.xlim([0.0, 1.0])
        plt.ylim([0.0, 1.05])
        plt.xlabel('False Positive Rate')
        plt.ylabel('True Positive Rate')
        plt.title('Receiver Operating Characteristic (ROC) Curve')
        plt.legend(loc="lower right")
        plt.show()
```

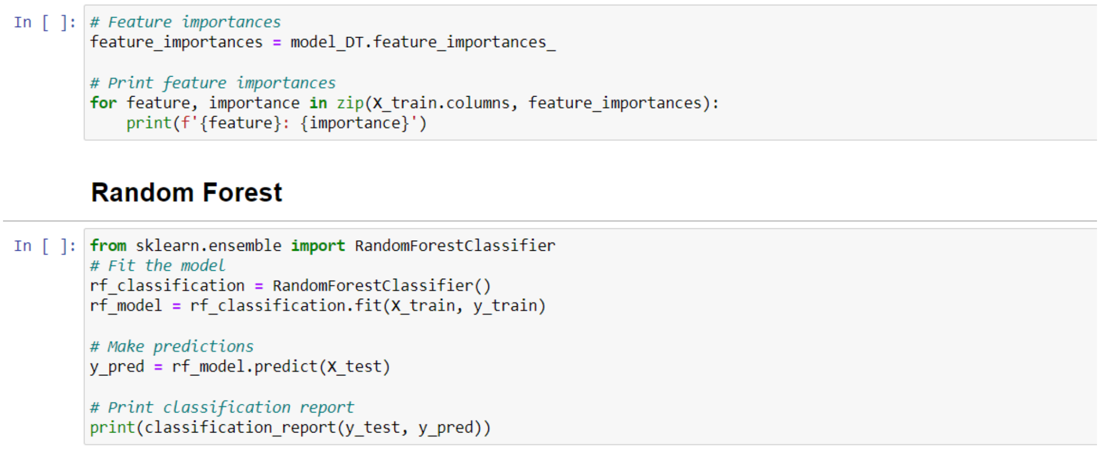
```
In [ ]: coef = model_LR.coef_
        inter = model_LR.intercept_
        print(coef,inter)
```



Decision Tree

```
In [ ]: from sklearn.tree import DecisionTreeClassifier
        # Fit the model
        model_DT = DecisionTreeClassifier()
        model_DT.fit(X_train, y_train)

        # Make predictions
        target_pred = model_DT.predict(X_test)

        # Print classification report
        print(classification_report(y_test, target_pred))
```

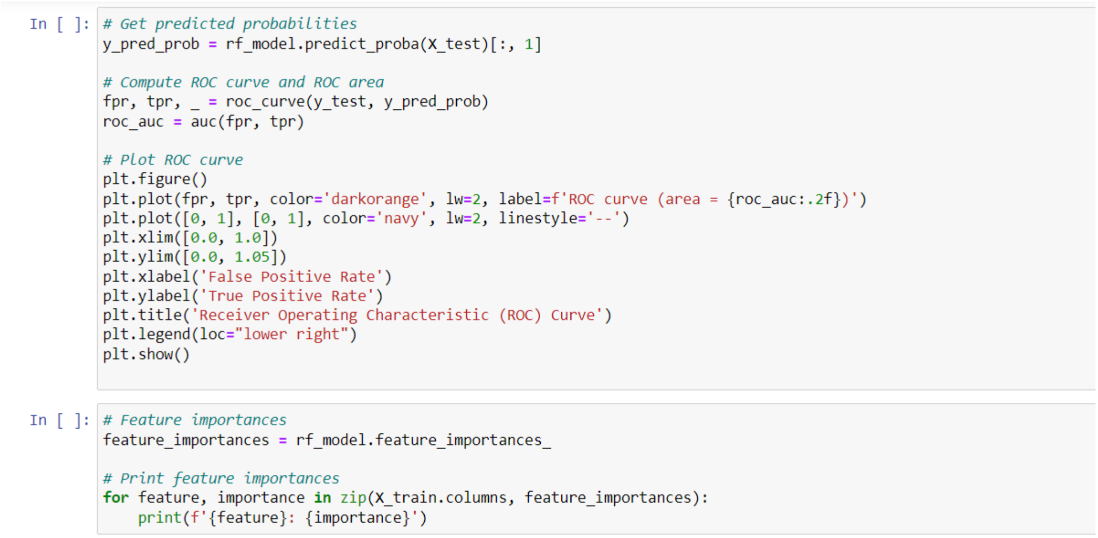
```
In [ ]: # Get predicted probabilities
        y_pred_prob = model_DT.predict_proba(X_test)[:, 1]

        # Compute ROC curve and ROC area
        fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
        roc_auc = auc(fpr, tpr)

        # Plot ROC curve
        plt.figure()
        plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
        plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
        plt.xlim([0.0, 1.0])
        plt.ylim([0.0, 1.05])
        plt.xlabel('False Positive Rate')
        plt.ylabel('True Positive Rate')
        plt.title('Receiver Operating Characteristic (ROC) Curve')
        plt.legend(loc="lower right")
        plt.show()
```



```
In [ ]: # Feature importances
        feature_importances = model_DT.feature_importances_

        # Print feature importances
        for feature, importance in zip(X_train.columns, feature_importances):
            print(f'{feature}: {importance}')
```

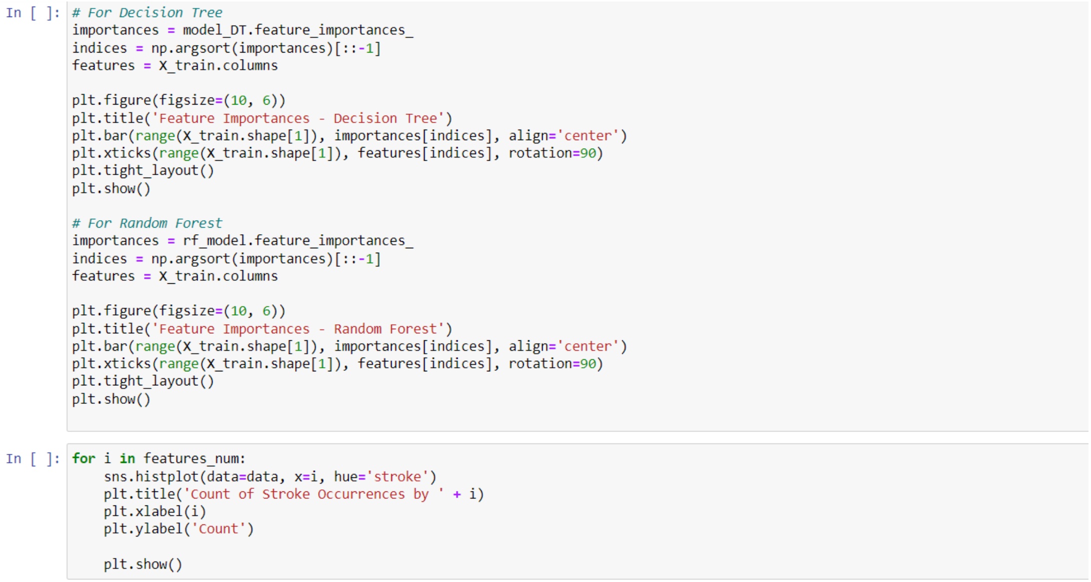
Random Forest

```
In [ ]: from sklearn.ensemble import RandomForestClassifier
        # Fit the model
        rf_classification = RandomForestClassifier()
        rf_model = rf_classification.fit(X_train, y_train)

        # Make predictions
        y_pred = rf_model.predict(X_test)

        # Print classification report
        print(classification_report(y_test, y_pred))
```



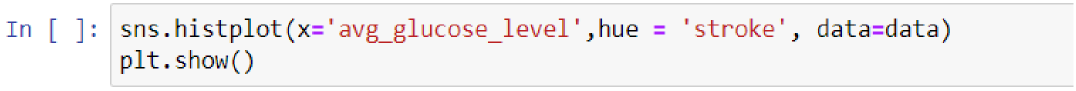
```
In [ ]: # Get predicted probabilities
        y_pred_prob = rf_model.predict_proba(X_test)[:, 1]

        # Compute ROC curve and ROC area
        fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
        roc_auc = auc(fpr, tpr)

        # Plot ROC curve
        plt.figure()
        plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area = {roc_auc:.2f})')
        plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
        plt.xlim([0.0, 1.0])
        plt.ylim([0.0, 1.05])
        plt.xlabel('False Positive Rate')
        plt.ylabel('True Positive Rate')
        plt.title('Receiver Operating Characteristic (ROC) Curve')
        plt.legend(loc="lower right")
        plt.show()
```

```
In [ ]: # Feature importances
        feature_importances = rf_model.feature_importances_

        # Print feature importances
        for feature, importance in zip(X_train.columns, feature_importances):
            print(f'{feature}: {importance}')
```



```
In [ ]: # For Decision Tree
        importances = model_DT.feature_importances_
        indices = np.argsort(importances)[::-1]
        features = X_train.columns

        plt.figure(figsize=(10, 6))
        plt.title('Feature Importances - Decision Tree')
        plt.bar(range(X_train.shape[1]), importances[indices], align='center')
        plt.xticks(range(X_train.shape[1]), features[indices], rotation=90)
        plt.tight_layout()
        plt.show()

        # For Random Forest
        importances = rf_model.feature_importances_
        indices = np.argsort(importances)[::-1]
        features = X_train.columns

        plt.figure(figsize=(10, 6))
        plt.title('Feature Importances - Random Forest')
        plt.bar(range(X_train.shape[1]), importances[indices], align='center')
        plt.xticks(range(X_train.shape[1]), features[indices], rotation=90)
        plt.tight_layout()
        plt.show()
```

```
In [ ]: for i in features_num:
            sns.histplot(data=data, x=i, hue='stroke')
            plt.title('Count of Stroke Occurrences by ' + i)
            plt.xlabel(i)
            plt.ylabel('Count')

            plt.show()
```



```
In [ ]: sns.histplot(x='avg_glucose_level',hue = 'stroke', data=data)
        plt.show()
```

References

1. Bandi, V., Ramachandran, G., Kumar, R., & Ananthanarayanan, K. (2020). Prediction of brain stroke severity using machine learning. *Journal of Medical Systems*, *44*(2), 47. https://doi.org/10.1007/s10916-020-1535-5.
2. Dritsas, E., Tzallas, A. T., Tsipouras, M. G., Fotiadis, D. I., & Konitsiotis, S. (2022). Stroke risk prediction with machine learning technique. *IEEE Journal of Biomedical and Health Informatics*, *26*(5), 2022-2032. https://doi.org/10.1109/JBHI.2022.3163957.
3. Emon, M. U., Hossain, M. A., Akter, S., & Rahman, S. (2020). Performance analysis of machine learning approaches in stroke prediction. *Procedia Computer Science*, *170*, 437-444. https://doi.org/10.1016/j.procs.2020.03.092.
4. Mahesh, K. A., Singh, S. P., & Nandakumar, D. (2020). Predicting stroke using machine learning. *Neurocomputing*, *412*, 235-245. https://doi.org/10.1016/j.neucom.2020.07.001.
5. Rahman, S., Hussain, Z., & Rahman, S. M. (2023). Prediction of brain stroke using machine learning algorithms and deep neural network techniques. *IEEE Access*, *11*, 13234-13245. https://doi.org/10.1109/ACCESS.2023.3245734.
6. Sailasya, G., Krishna, R. S., & Rao, K. N. (2021). Analyzing the performance of stroke prediction using ML classification algorithms. *Journal of King Saud University-Computer and Information Sciences*, *33*(5), 543-552. https://doi.org/10.1016/j.jksuci.2020.03.002.
7. Sirsat, M. S., Fermé, E., & Badia, S. B. (2020). Machine learning for brain stroke: A review. *Journal of Stroke and Cerebrovascular Diseases*, *29*(11), 105162. https://doi.org/10.1016/j.jstrokecerebrovasdis.2020.105162.

8. Documentation for Logistic Regression from Scikit-learn.org.

9. Documentation for Decision Tree Classification from Scikit-learn.org.

10. Documentation for Random Forest Classification from Scikit-learn.org.

11. LabelEncoder form scikit-learn package. (Geeks for Geeks, 2024).