Assignment: 2

Classification

Stroke Prediction Models

Instructor: Mr. Bisrat (Msc.)

Addis Ababa Institute of Technology

School of Information Technology and Engineering

Software Stream

Name	ID Number
Lidiya Mamo	UGR/2485/14
Nathan Mesfin	UGR/0534/14

TABLE OF CONTENTS

INTRODUCTION

Stroke is a medical emergency caused because of the loss of blood flow to part of the brain resulting in damage in brain tissue. It is one of the leading causes of death and disability worldwide. According to the world health organization (WHO) approximately 15 million people suffer strokes each year. About 5 million of these cases are fatal and another 5 million cause permanent disability. Predicting stroke is crucial for enabling timely interventions by reducing morbidity and mortality rates as a result.

In this project, we will apply machine learning techniques to classify individuals based on their likelihood of experiencing stroke. The input features include: gender, age, past diseases, smoking status and the like. This report evaluates and compares six well-known classification algorithms namely: Logistic Regression, Naive Bayes, Gaussian Discriminant Analysis (GDA), Support Vector Machine (SVM), Decision Tree, and Random Forest. By doing so, we aim to identify the most effective model for the data distribution in use.

DATA PREPROCESSING AND FEATURE ENGINEERING

About the dataset

We utilized the Stroke Prediction dataset from Kaggle for this experiment *(link to the dataset is provided in the reference section).* The dataset includes a multitude of features relevant to stroke prediction. Some of these were related to the demographics, health metrics and lifestyle of individuals. Before the cleaning and filtering process, these were the columns in the dataset:

- **Demographics**: id, gender, age, residence type
- **Health Metrics**: hypertension, heart disease, average glucose level, BMI
- **Lifestyle Factors**: ever married, work type, smoking status
- **Target Variable**: stroke (binary: 1 for stroke, 0 otherwise)

Preprocessing Steps

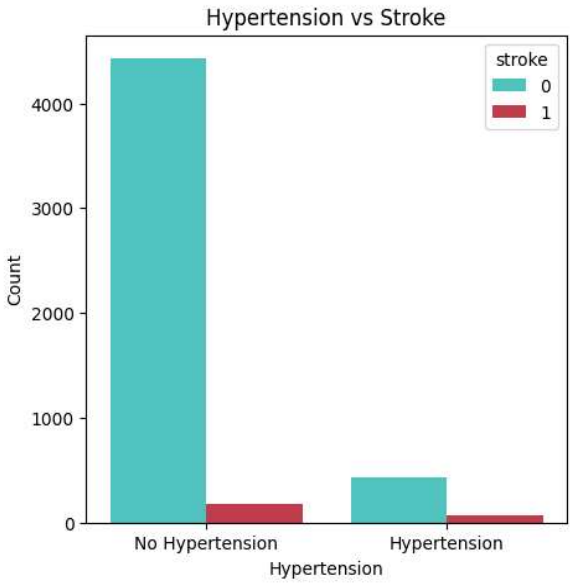
1. There were 201 missing values in the *BMI* column. These missing values were imputed with the column's means to avoid dropping rows altogether so that the dataset remained complete for analysis.
2. The *id* column was dropped since it presented no significance for our purposes.
3. We used *Label Encoder* to encode categorical features such as: *gender, ever_married, work_type, Residence_type,* and *smoking_status.* This technique assigns unique integer

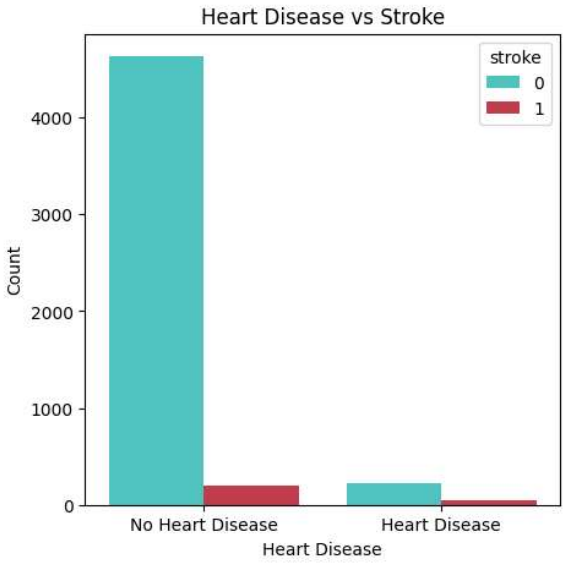
values to each category in the respective columns. This enables compatibility with machine learning algorithms.

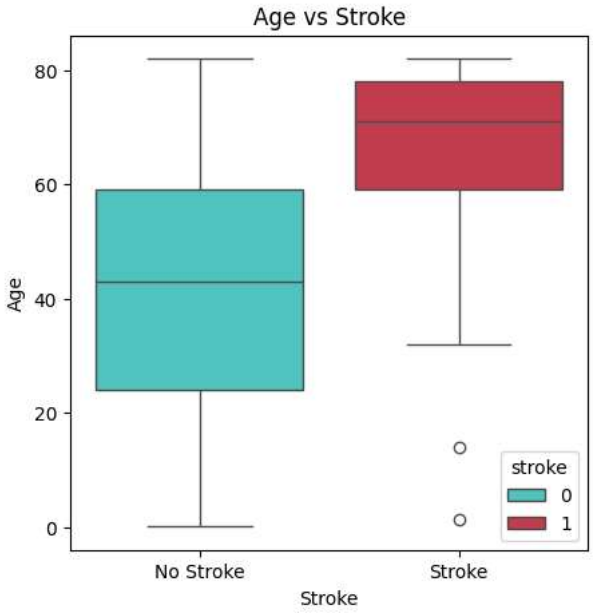
4. We also standardized continuous variables (*age, avg_glucose_level,* and *bmi)* using *StandardScaler.* With this, we ensured that the mean of these features was 0 and their standard deviation was 1. By doing so, they were made comparable in magnitude. This has various benefits for the performance of the algorithms used in this project. Some of which include:
    i. Eliminating feature bias. Since models that calculate distances or coefficients like SVM and Logistic regression can be biased towards features with larger scale.
    ii. Faster convergence of optimization-based models like gradient ascent in logistic regression.
5. The dataset was split into training and test sets in a 70:30 ratio.
6. We used the SMOTE(*Synthetic Minority Oversampling Technique*) to address class imbalance in the target variable. SMOTE generated samples for the minority class *(stroke = 1)* in the training set. This prevents overfitting by ensuring balanced distribution of classes for the models.

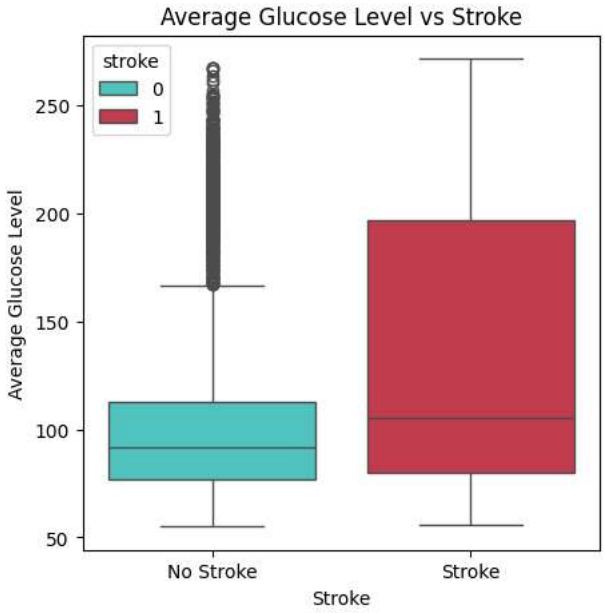
Visualizing the Distribution

Visualization of the relationship between the target and the independent variables revealed interesting patterns, such as higher stroke prevalence in older adults and individuals with heart disease or hypertension.









See Appendix A and B for feature distribution visualizations. Additional visualizations for the relationship between the target and the rest of the independent variables are also listed in the appendices

Libraries Used

We used the following python libraries for the project:

- **Data Processing:** pandas
- **Pandas:** For data processing
- **Matplotlib & Seaborn:** For data visualization
- **Scikit-Learn:** For modeling and machine learning
- **Imbalanced-Learn:** for handling imbalanced data.

TRAINING THE MODELS

Logistic Regression

Logistic regression is a common classification algorithm. It models the probability of the target as a function of the input features. It uses the logistic (also known as sigmoid) function to map predictions to probabilities between 0 and 1. In our context, the algorithm predicts likelihood of stroke based on the linear relationships between features such as age, glucose levels and body mass index (BMI) which is then passed through the sigmoid function to produce probability.

Naive Bayes

This is a probabilistic classifier that is built on the bayes theorem. Naïve Bayes presupposes the independence of predictors. Although it is simple, the algorithm performs quite well in text and medical classification tasks which is what prompted us to use it for this project. Here, the algorithm estimates the probability of stroke based on the conditional probabilities of features such as hypertension, smoking status and the like.

Gaussian Discriminant Analysis (GDA)

GDA is another algorithm used in this project. It assumes the data from each class follows a Normal (Gaussian) distribution. It models the relationship between the features by estimating class specific means and utilizing a shared covariance matrix. This gives the algorithm the capability to handle overlapping data distributions in an effective manner. In this context, the algorithm will create separate models for the positive and negative class and categorize new data points based on their resemblance to one of the two classes.

Support Vector Machine (SVM)

SVM works by constructing a hyperplane that best separates the data points of the different classes. SVM can capture non-linear relationships in the data by making use of a radial basis function (RBF) kernel. For this project, we have used SVM to identify the complex boundaries between the stroke and non-stroke classes.

Decision Tree

This algorithm is one of the two additional algorithms that we added for this project. It is quite commonly used for medical classification purposes. Decision trees work by partitioning the data into subsets based on feature thresholds. This creates a tree-like structure for

classification. Here, Decision Tree will build a model by recursively splitting the data at thresholds for each feature and predict the likelihood of stroke based on the learned patterns.

Random Forest

To take the decision of trees a step further, we also included the Random Forest algorithm. This work much like decision trees but instead of having a single tree which is often prone to overfitting, it instead make use of several decision trees. Each tree will be exposed to different parts of the data and in the process; the model understands the complex relationships present in the distribution. The trees will each make their own class predictions based on the features they are using and random forest takes the majority vote to make its prediction. In this project, Random Forest uses multiple decision trees to predict whether a given datapoint belongs to the stroke or the non-stroke classes.

PERFORMANCE EVALUATION

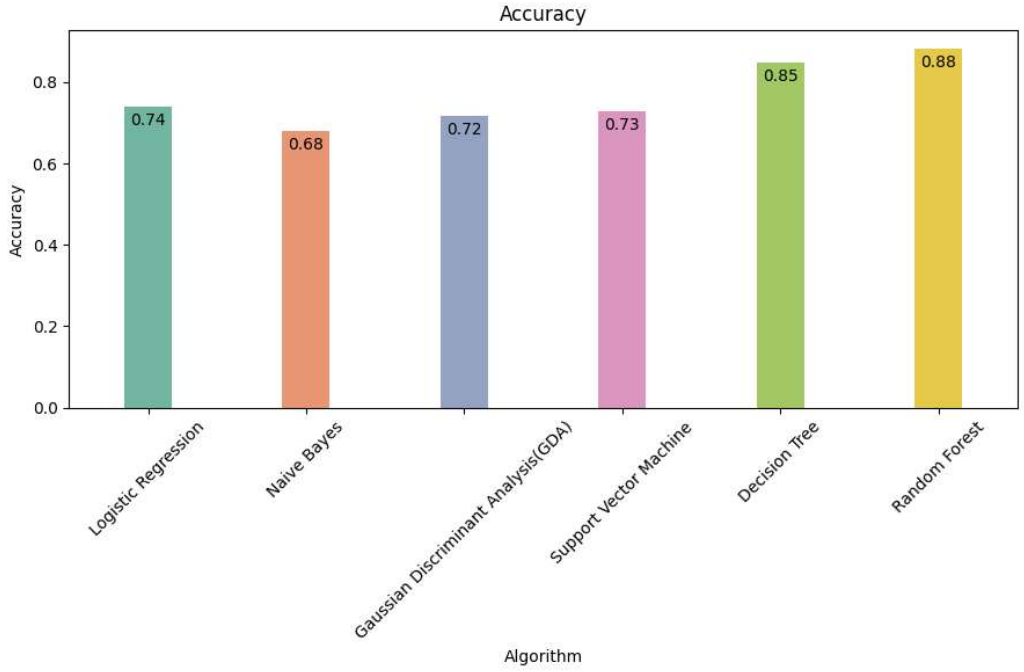
The following metrics were used to evaluate the performance of the models:

- **Accuracy**:

  This is perhaps the simplest evaluation metric. It measures the proportion of correct predictions made from all predictions.

$$Accuracy = \frac{Correct\ predictions}{Total\ predictions}$$

  It is important to note that it may not fully reflect the performance in imbalanced datasets. During our pre-processing step we used SMOTE to handle the imbalance caused due to the fact that non-stroke cases are more frequent than stroke cases. Nevertheless, accuracy alone will not be used as an evaluation metric in this report. The following is a graphical representation of the accuracy of the predictions across the 6 classifiers. Showing that the “Random Forest” model is the most accurate as it provides a prediction that is accurate 88% of the time. And the Naïve bayes model is the least accurate with a prediction accuracy of 68%. This is understandable since the nature of the algorithm entails it makes assumptions that can be misleading.

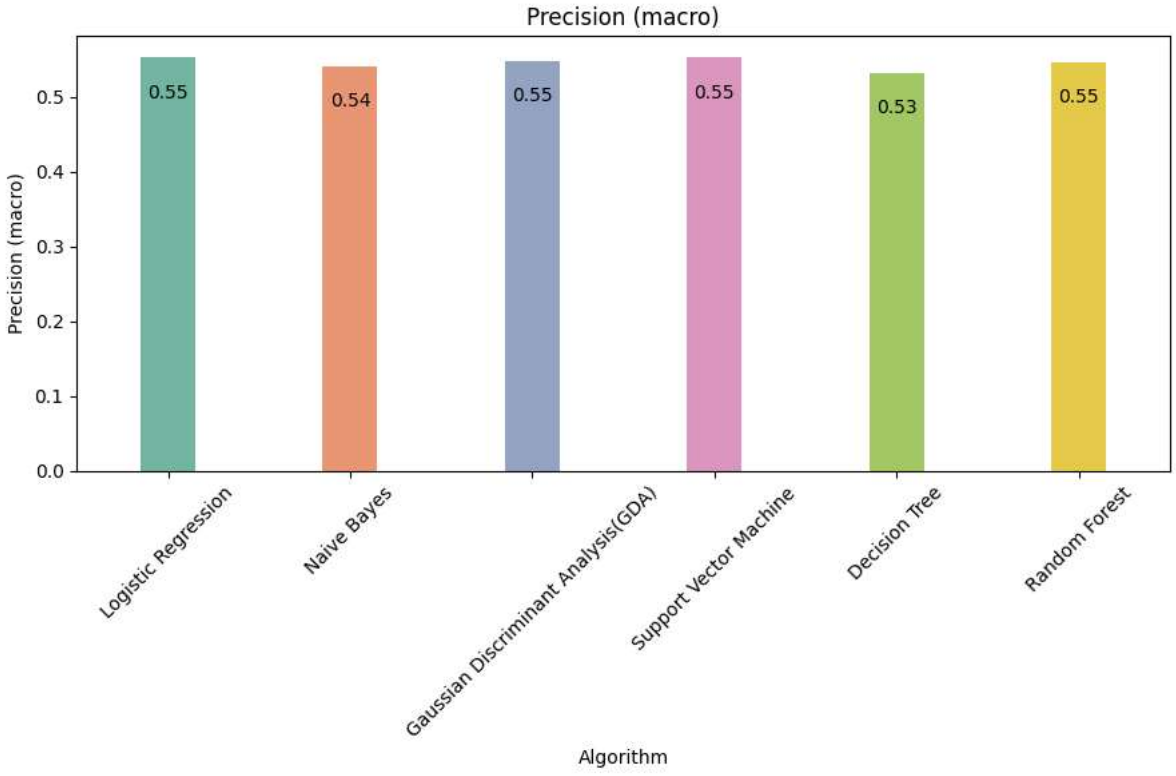


❖ **Precision(Macro):**

Precision tells us just how many of the positive predictions made by the model are correct. Or in other words we are asking, "Of all the times the models gave us a stroke positive prediction, how many of those actually turned out to be strokes?" It is calculated by using the following equation:

$$Precision = \frac{True\ Positives}{True\ Positives + Fals\ \ Positives}$$

The following graph compares the precision of the 6 models:

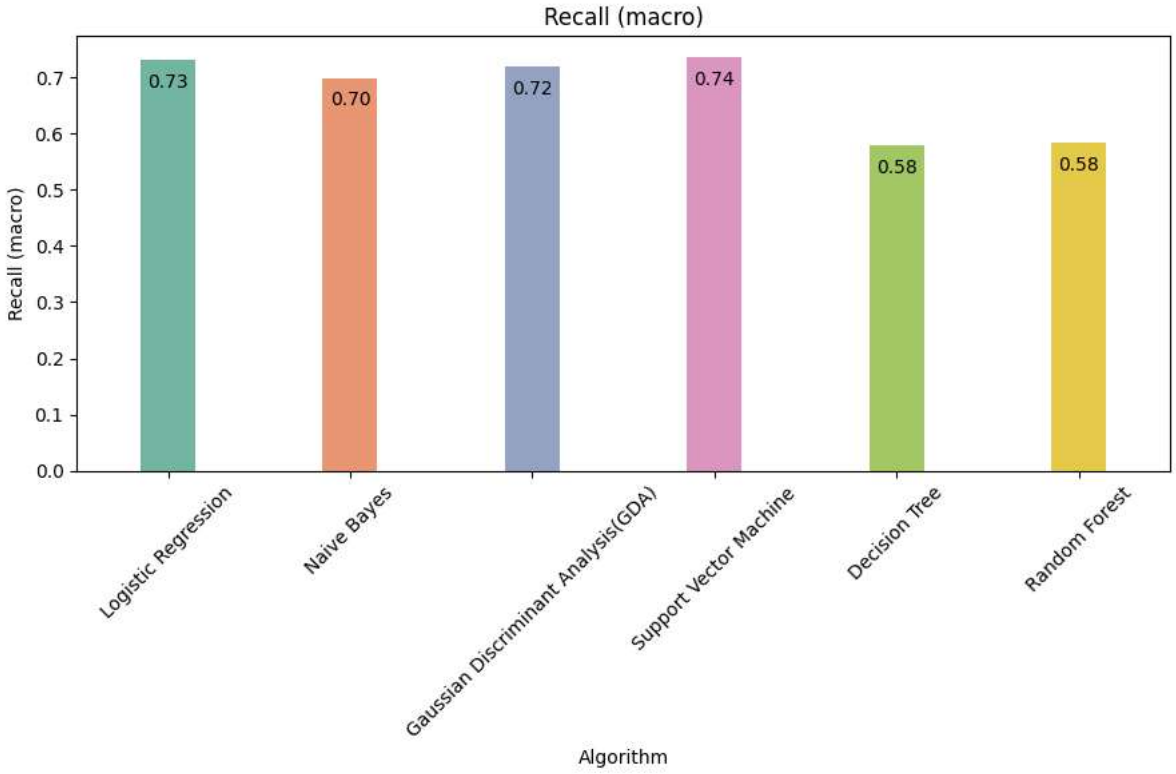


Here we observe that the precision is similar across Logistic regression, SVM, GDA and Random Forest. We also observe that the Decision Tree model has slightly less precision compared to the other models.

- **Recall (Macro)**:

  This metric measures how well the model captures actual positive cases i.e true strokes. We are asking "How many of the true strokes were correctly identified by the models?"

$$Recall = \frac{True\ Positives}{True\ Positives + Fa \quad Negatives}$$



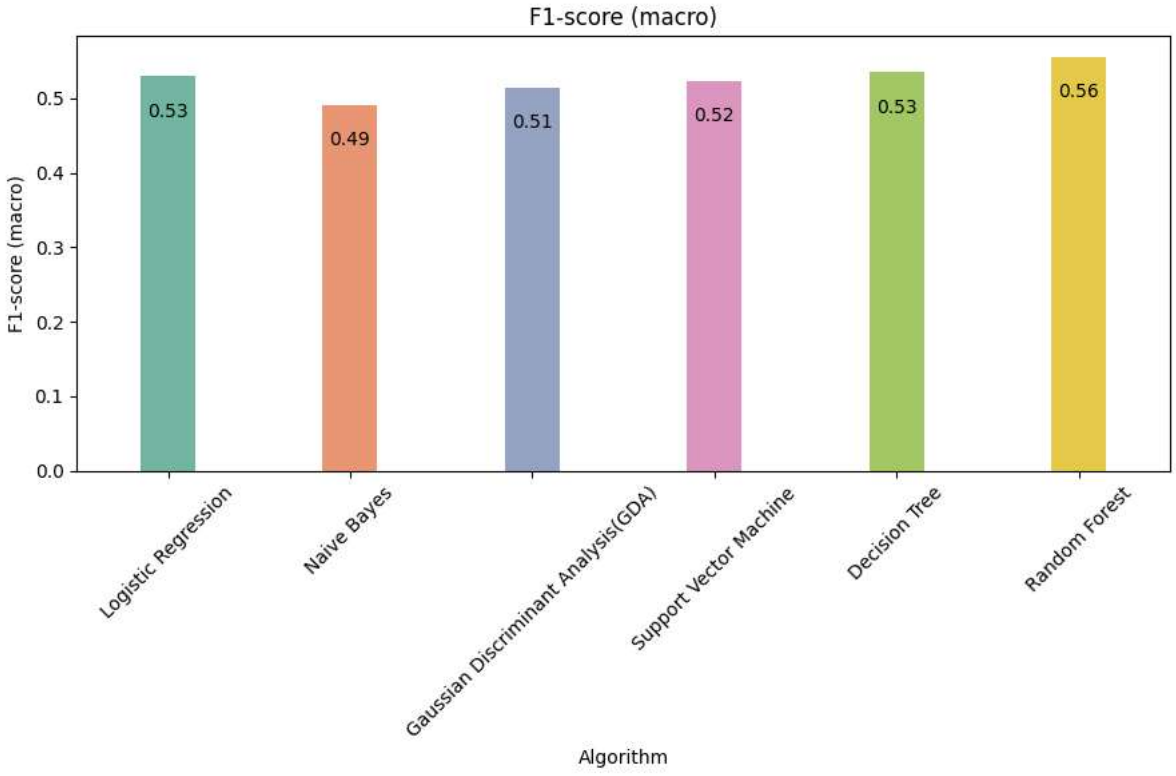
Observing the graph provided above indicates SVM has much better recall compared to the rest of the models - closely followed by GDA.

- **F1-Score (Macro):**

This metric helps us balance precision and recall. It presents the harmonic mean of precision and recall and presents us with a single value that reflects both the precision and the recall of the models. Mathematically F1 score is calculated by:

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

Based on our evaluation, the Random Forest model gave a higher F1 score value compared to the rest of the models. Below, is a graph showing the result for F1 scores across the 6 models:



The following table summarizes our findings for the performance evaluation metrics across the 6 models:

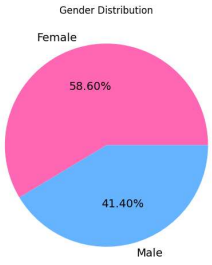
Algorithm	*Accuracy*	*Precision (Macro)*	*Recall (Macro)*	*F1-Score (Macro)*
Logistic Regression	74.10%	55.33%	73.11%	52.94%
Naive Bayes	68.10%	54.13%	69.95%	49.14%
Gaussian Discriminant Analysis (GDA)	71.75%	54.81%	71.87%	51.41%
Support Vector Machine	72.80%	55.29%	73.68%	52.37%
Decision Tree	84.80%	53.22%	57.86%	53.46%
Random Forest	88.26%	54.63%	58.41%	55.56%

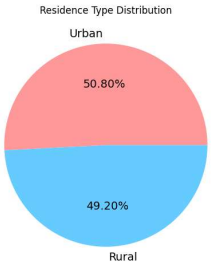
CONCLUSION

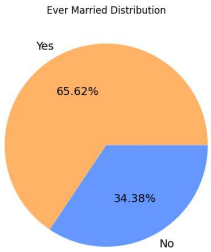
In this project, we have built 6 different models to predict the likelihood of an individual getting stroke. Our insight into the performance evaluation has shown that the Random Forest model outperformed all other models under nearly all performance evaluation metrics. Indicating it is the best model for data distribution. We assume that the superiority of this model is likely due to the ensemble approach it uses to reduce overfitting. The SVM and Logistic regression models have also performed reasonably well. We were also able to demonstrate the importance of feature scaling, handling class imbalance and selecting proper evaluation metrics for healthcare predictive modeling.

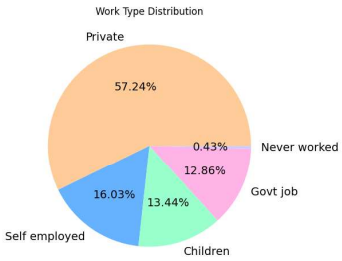
APPENDIX

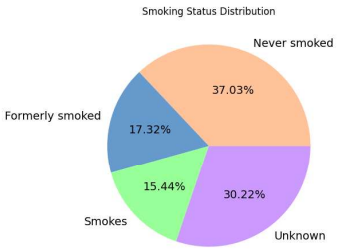
A. Visual representation of Categorical features

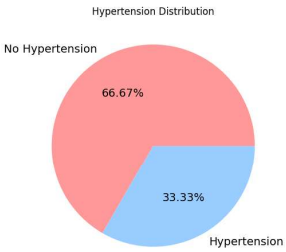


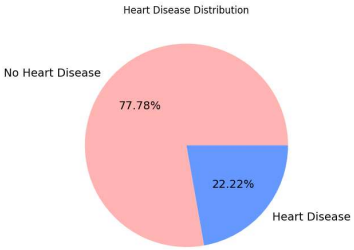




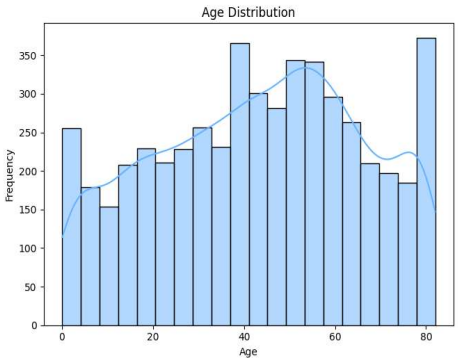


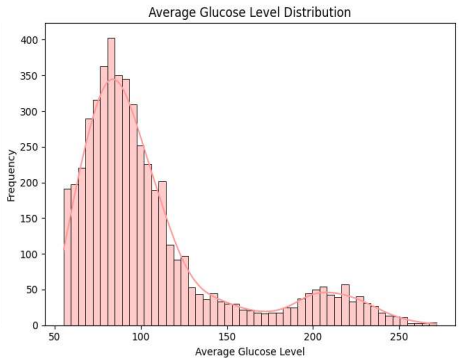


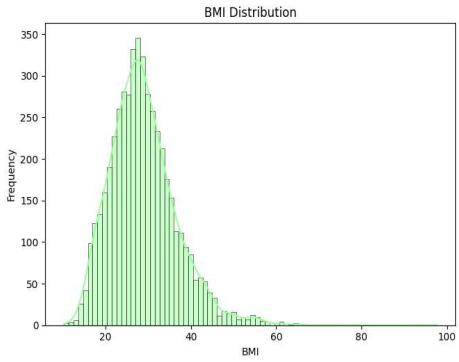




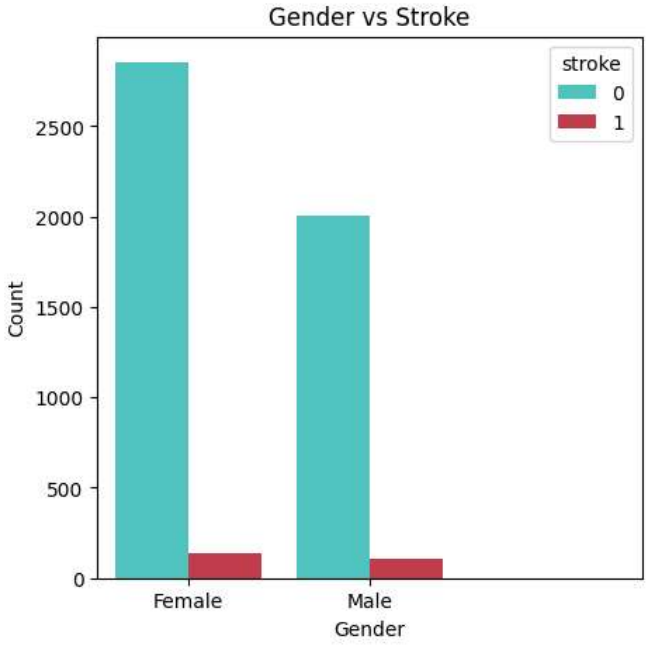
B. Visual representation of Numerical features



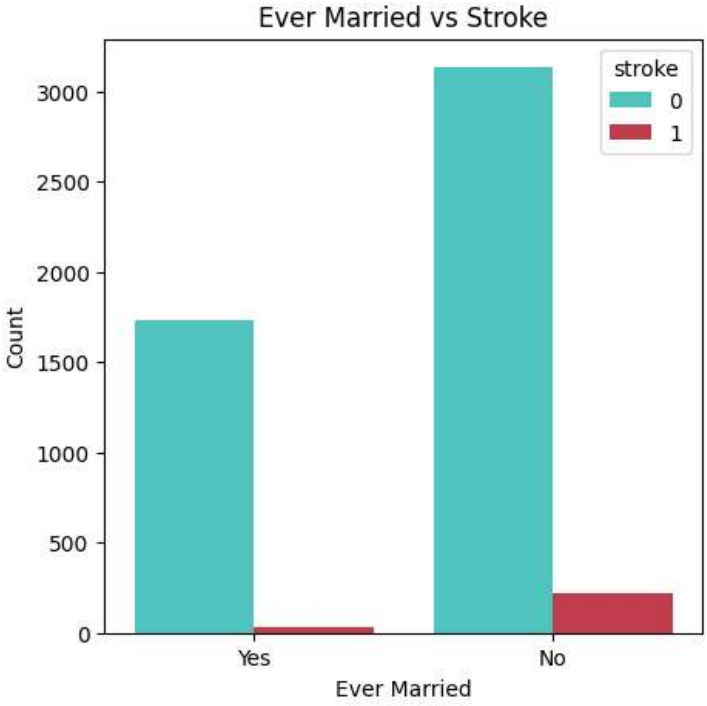




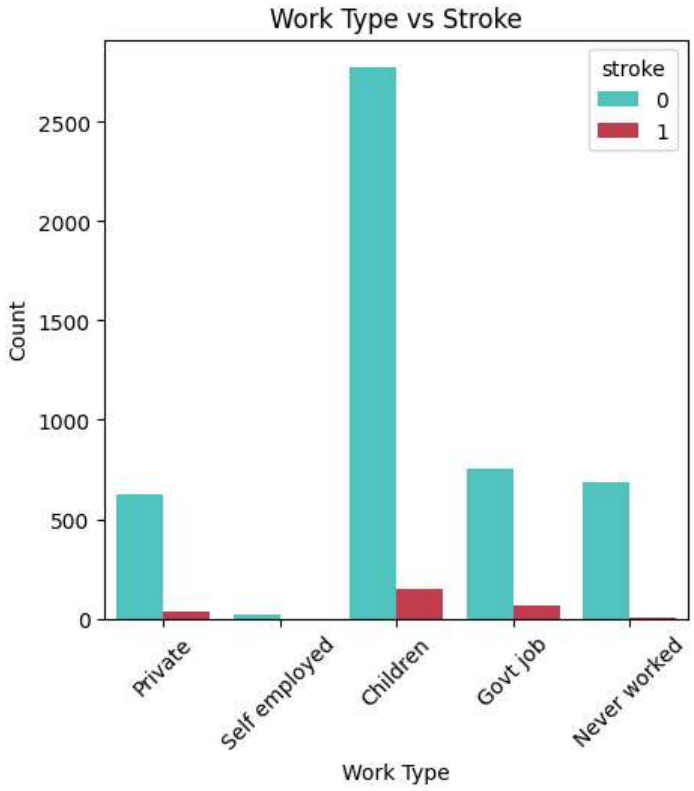
C. Gender Vs Stroke



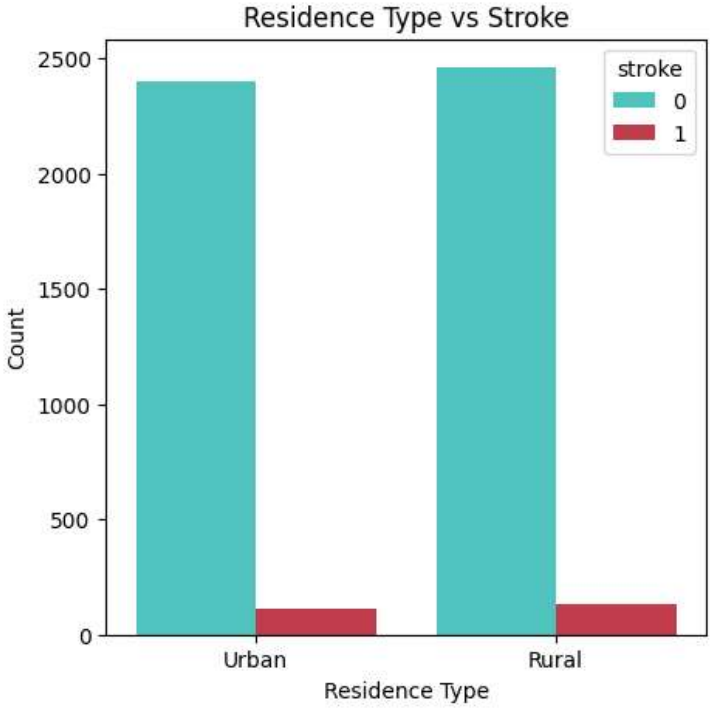
D. Marriage Vs Stroke



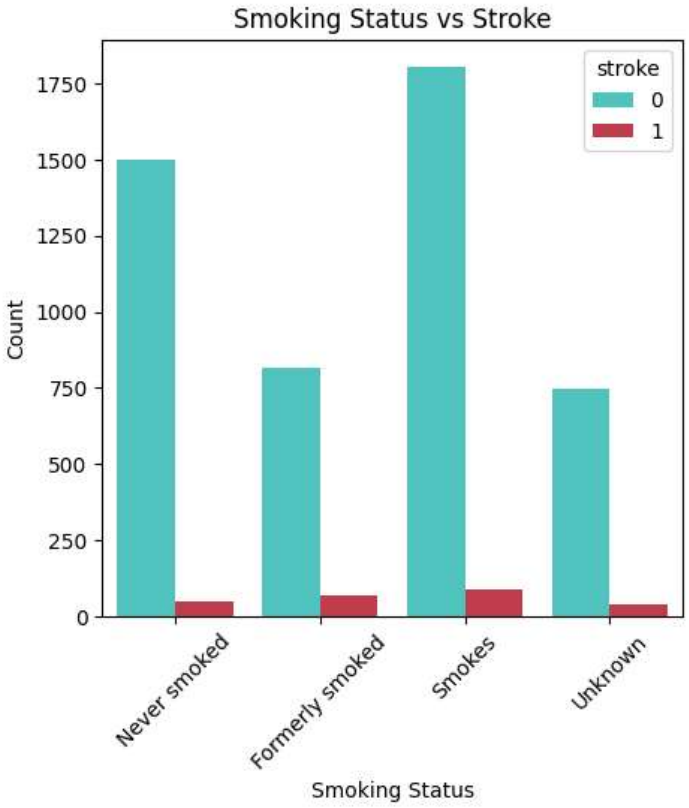
E. Work Type Vs Stroke



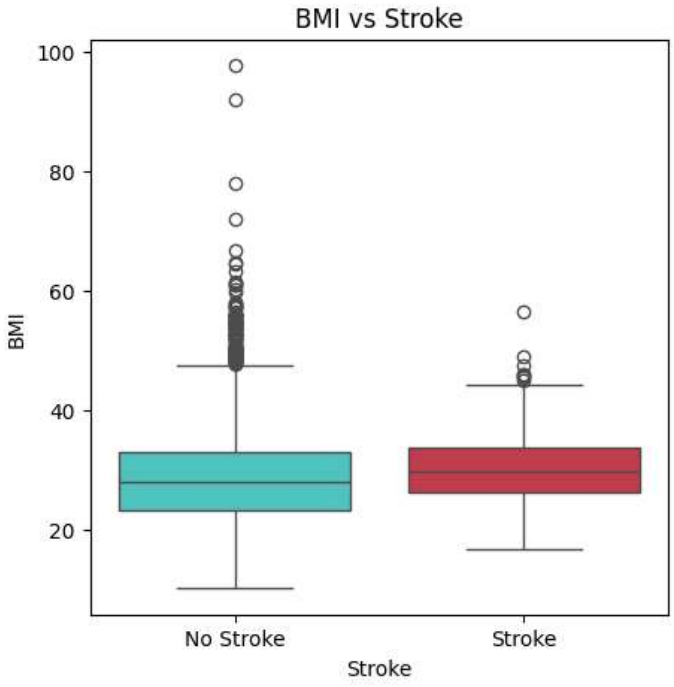
F. Residence Type Vs Stroke



G. Smoking Status Vs Stroke



H. BMI Vs Stroke



REFERENCES

- **Bishop, Christopher M.** (2006). *Pattern Recognition and Machine Learning. New York: Springer*
- **Murphy, K. P.** (2012). *Machine learning: a probabilistic perspective*

Web Sources:

- **Geek for Geeks:** *Random Forest Algorithm in Machine Learning*
- **Kaggle**: Stroke Prediction Dataset
- **Scikit Learn Website:** *Decision Trees*