Stroke Prediction Using Machine Learning

A project by aspiring Data Analysts...

- **Sanjeev Malik**
- **Pavan Aditya**
- **Sunil Kumar**
- **Deepitha**

Agenda

- Introduction
- Abstract
- Process / Flow
- Attribute Information
- Problem Statement
- Glimpse of the Data
- Problem Solution
- Data Description
- EDA
- Feature Engineering
- Machine Learning Alogrithms
- Accuracy Comparison
- Thank You

Abstract

Stroke is a medical emergency that occurs due to the interruption of flo of blood to a part of brain because of bleeding ot blood clots. Worldwide, it is the second major reason for deaths with an annual mortality rate of 5.5 million.
Every year more than 15 million people worldwide have a stroke, and in every 4 minutes, someone dies due to stroke.
A stroke is generally a consequence of a poor style of living and hence, preventable in upto 80% of cases. Therefore the prediction of stroke becomes necessary and should be used to prevent permanent damage by stroke. The currentwork predicted the stroke using the different machine learning models namely, Naive Bayes, Logistic Regression, SVM (Support Vector Machine), KNN, Random Forest Classifier, Decision Tree. The project presents the comparison aming all the machine learning algorithms. After the Analysis the results are shown individually at the end of the presentaion.

Back to Agenda Page

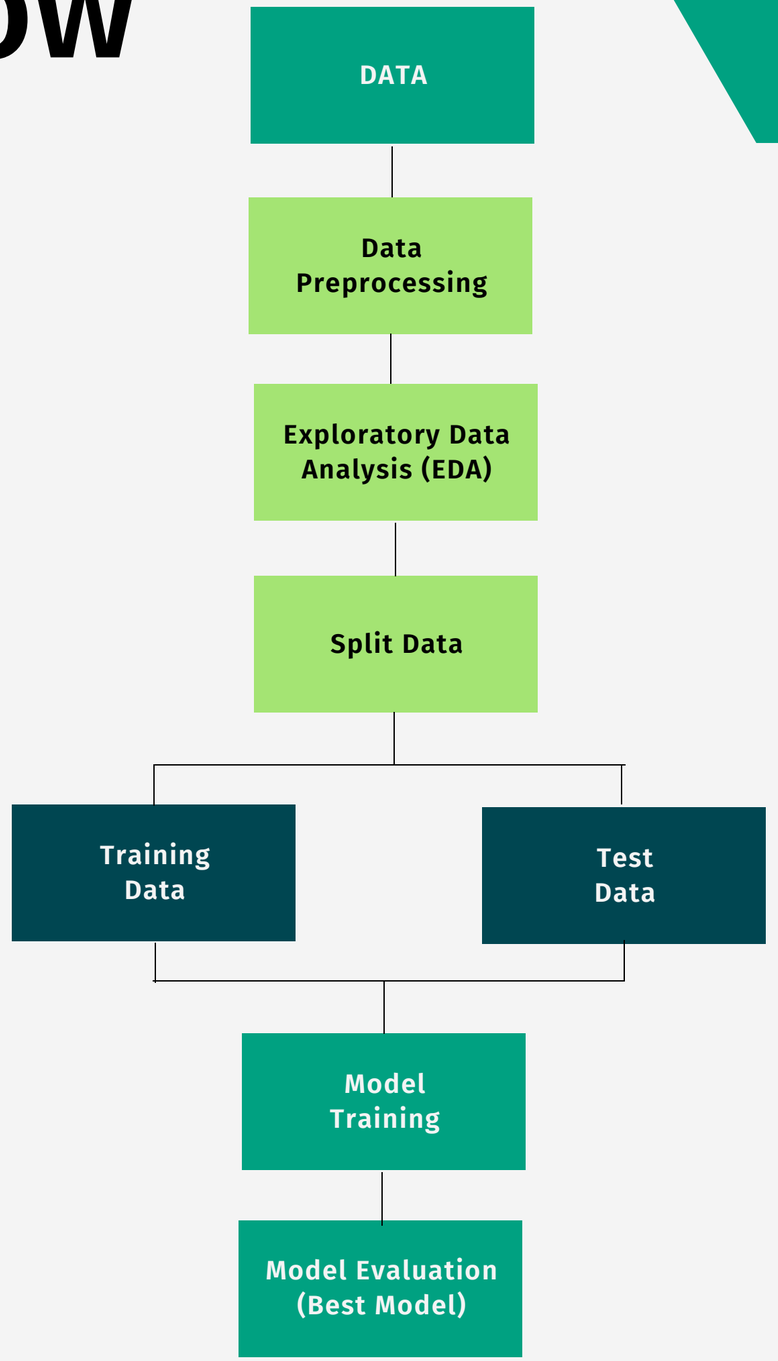
Introduction

A stroke happens when the blood flow to a part of the brain gets disrupted by blood clots or bleeding in the blood vessels, causing insufficient nutrients and oxygen in a part of the brain because of which the brain cells begin to die.
Stroke affects typically the blood vessels transporting the essential nutrients, blood, and oxygen to the brain. In order to prevent this damage and deaths immediate medical assistance is needed.

Back to Agenda Page

Process / Flow

Back to Agenda Page



Attribute Information:-

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

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

Problem Statement

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

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

Glimpse of the Data

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Existing System

A stroke is diagnosed by a person's symptoms, history, and lifestyle and is more likely to affect individuals exceeding the age of 55 years, suffering from heart diesases, uncontrolled high blood pressure, diabetes, obesity, consumes alcohol excessively, or smokes regularly.

A stroke is generally a result of a bad lifestyle and this can be prevented by making healthy lifestyle changes by avoiding alcohol or smoking and eating healthy food by maintaining a balanced diet.

Gaps in the Existing System

The main criteria is to identify the route cause and treat the issues.
So by implementing the various machine learning algorithms we can predict the stroke occurrence in a very efficient way will less cost.
This study helps us to predict the occurrence of the stroke by considering some specific key factors, conditions and lifestyle of a set of people.

Problem Solution

Machine Learning Algorithms

In this project our main aim is to bridge the gap by following a systematic analysis of the various patient records and their health history for the purpose of stroke prediction.

Here we have performed various machine learning algorithms such as K-Nearest (KNN), Support Vector Machine (SVM), Navie Bayes, Random Forest, and Decision Tree to make the prediction of the stroke.

Data Description

df.describe()

	id	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	4909.000000	4909.000000	4909.000000	4909.000000	4909.000000	4909.000000	4909.000000
mean	37064.313506	42.865374	0.091872	0.049501	105.305150	28.893237	0.042575
std	20995.098457	22.555115	0.288875	0.216934	44.424341	7.854067	0.201917
min	77.000000	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	18605.000000	25.000000	0.000000	0.000000	77.070000	23.500000	0.000000
50%	37608.000000	44.000000	0.000000	0.000000	91.680000	28.100000	0.000000
75%	55220.000000	60.000000	0.000000	0.000000	113.570000	33.100000	0.000000
max	72940.000000	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

The data set consists of 5110 columns and 12 rows.

Data Cleaning

```
id                   0
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

Null Values in bmi

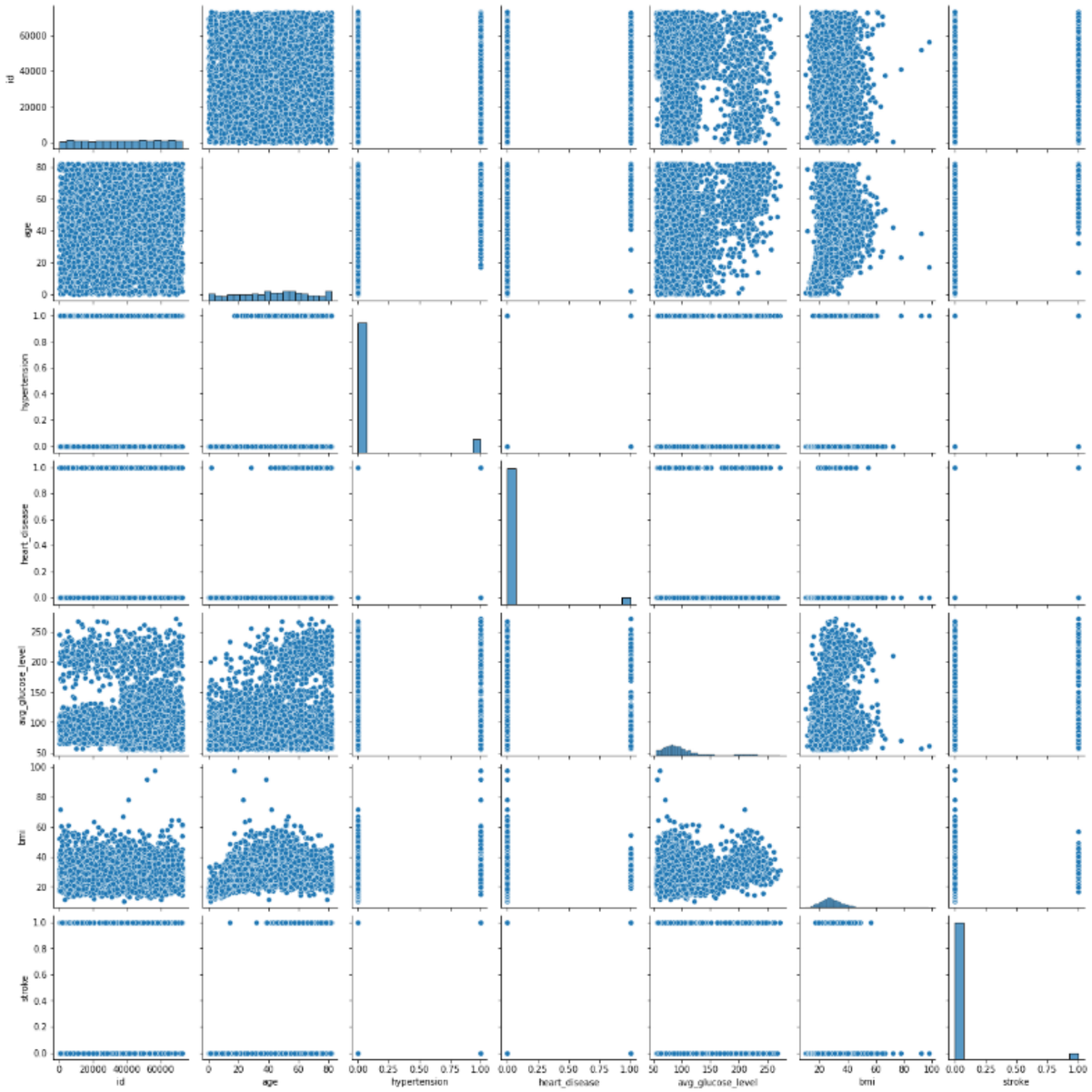
As we can there are 201 null values in bmi column which as to be treated.

```
id                   0
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
dtype: int64
```

Null Values are treated

In order to get rid of the null values we used a function df.dropna(inplace=True) this function helps in dropping of missing or null values.

Data Visulization



Data Visualization of cloumns using pairpot

EDA-Exploratory Data Analysis

- Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.
- EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate.

Categorical Variable

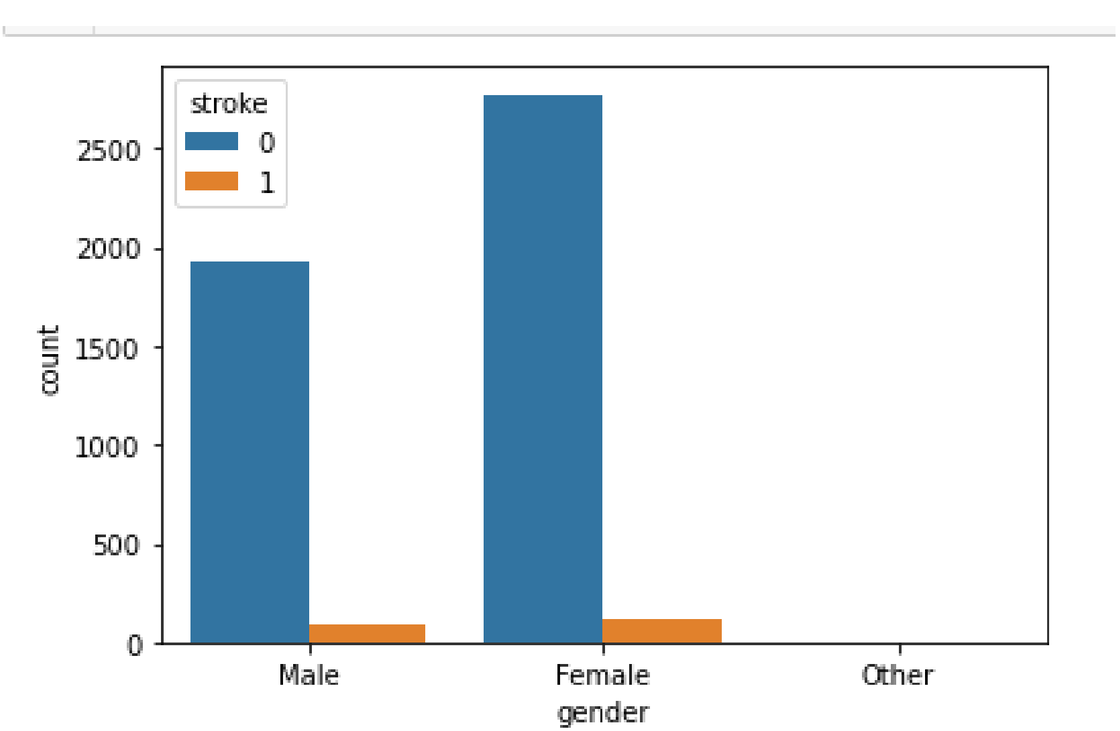
vs

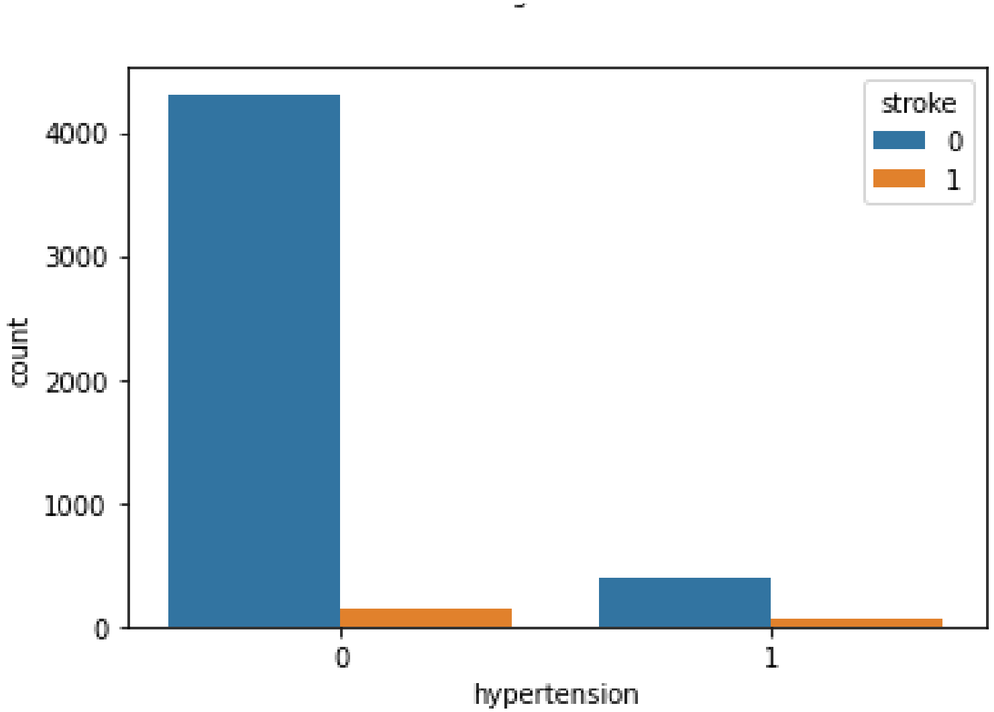
Conditional Variable

- Gender
- Hypertension
- Heart Diseases
- Ever Married
- Work Type
- Residence Type
- Smoking Status

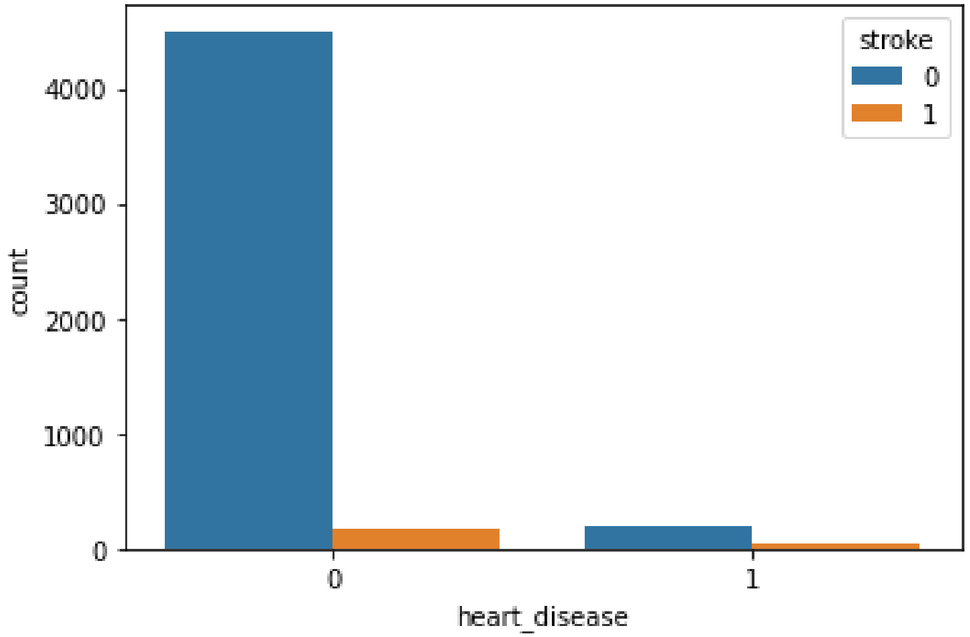
- Age
- Average Glucose Level
- Bmi

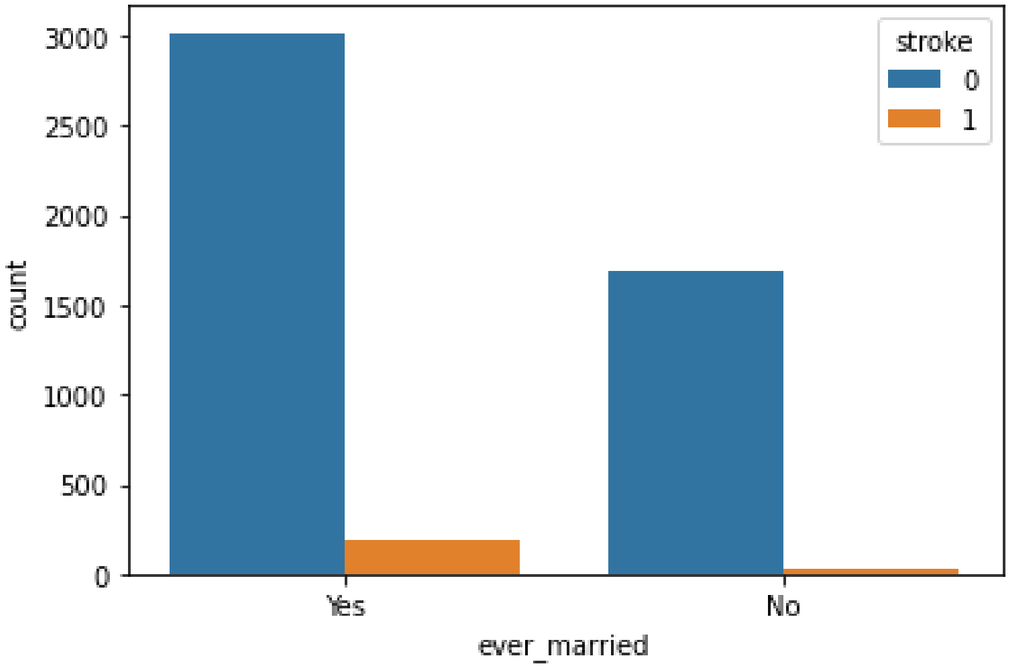
Categorical Variable



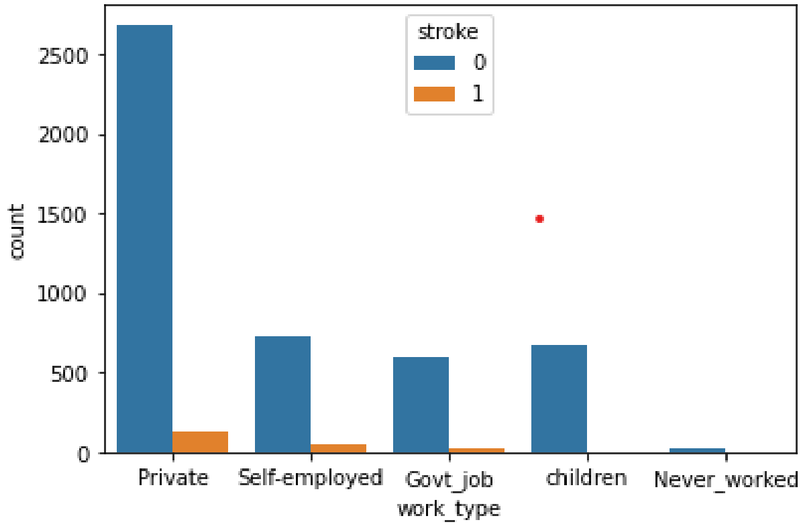


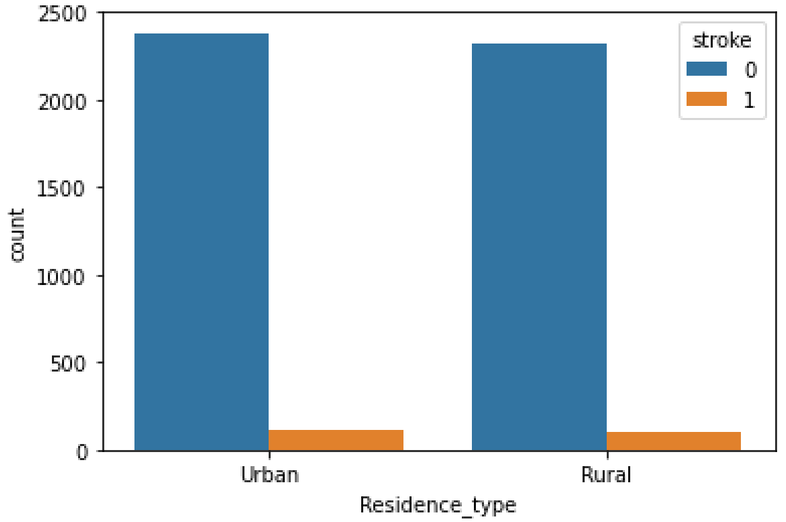
Categorical Variable

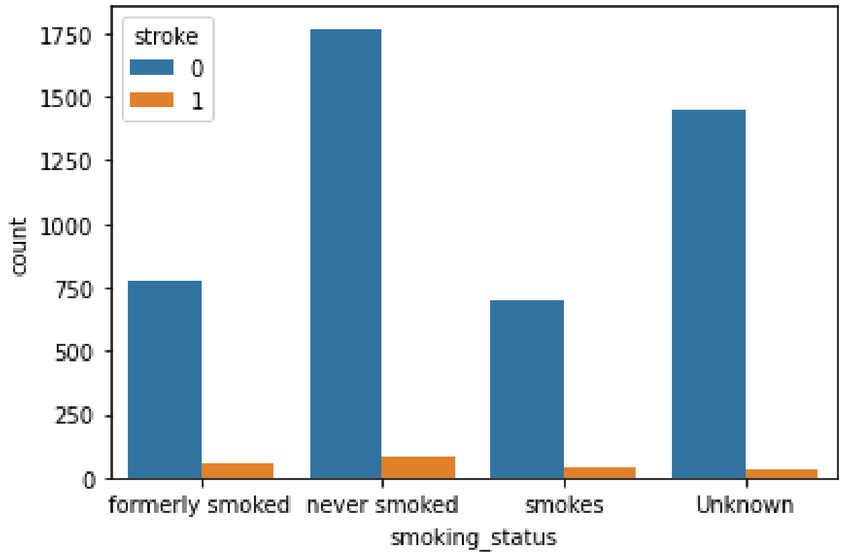




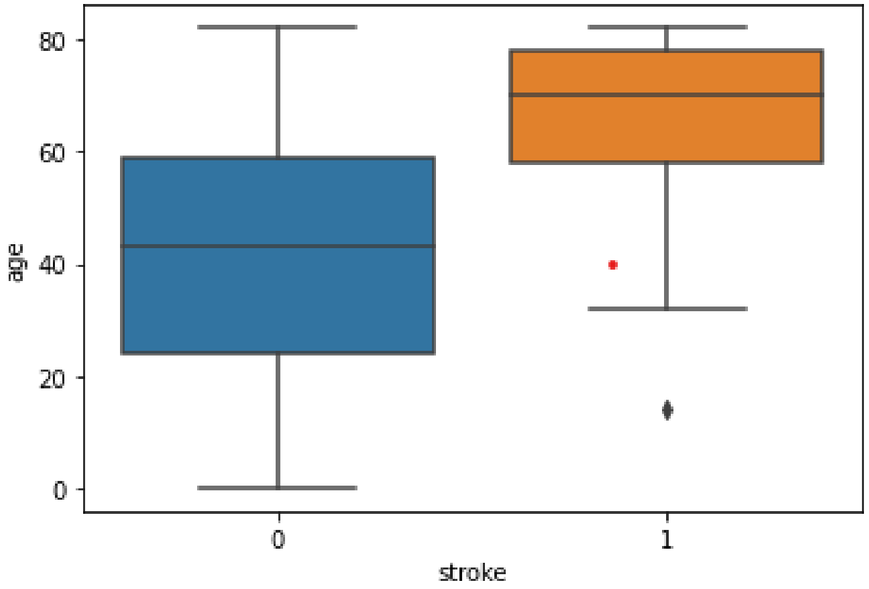
Categorical Variable

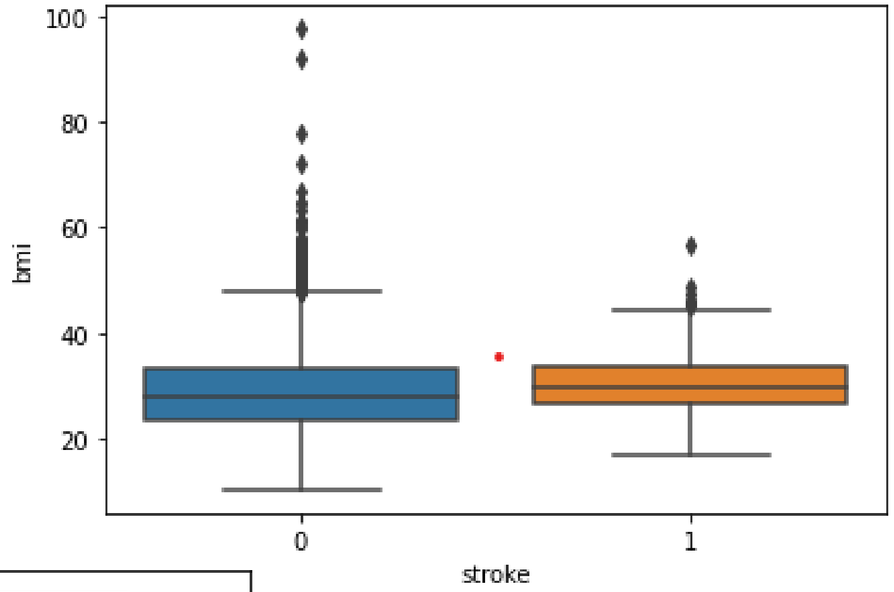


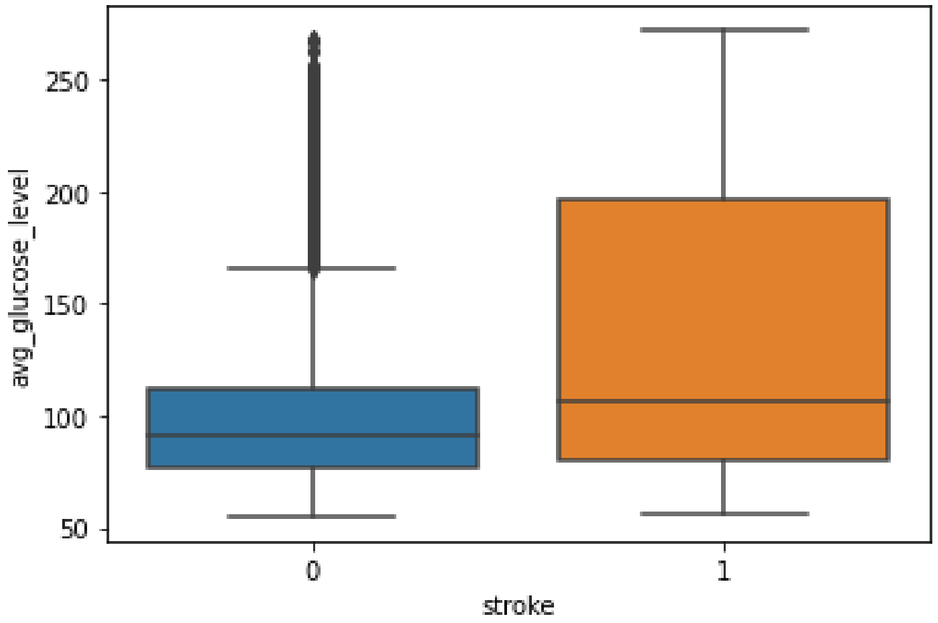




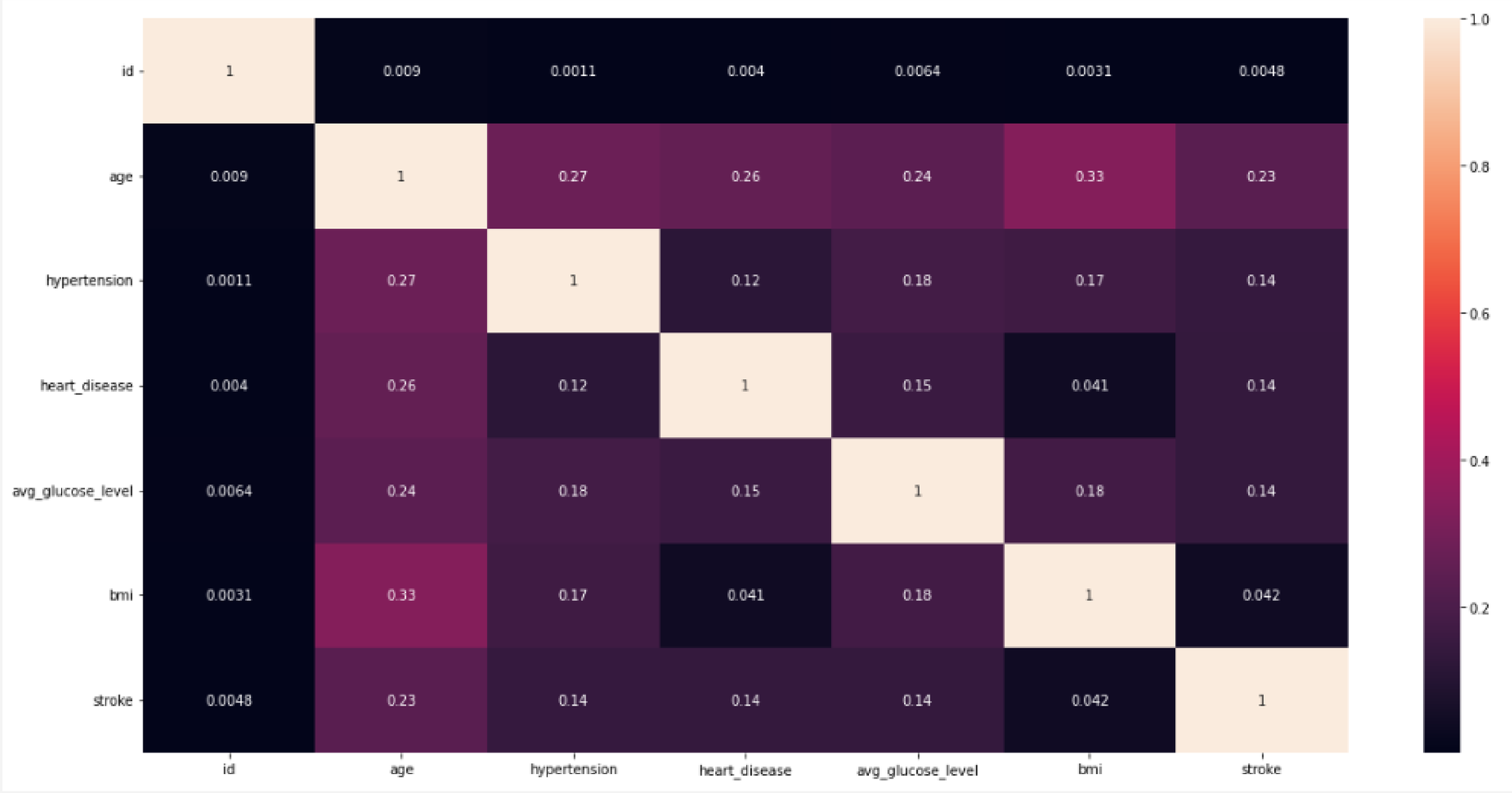
Continous Variable



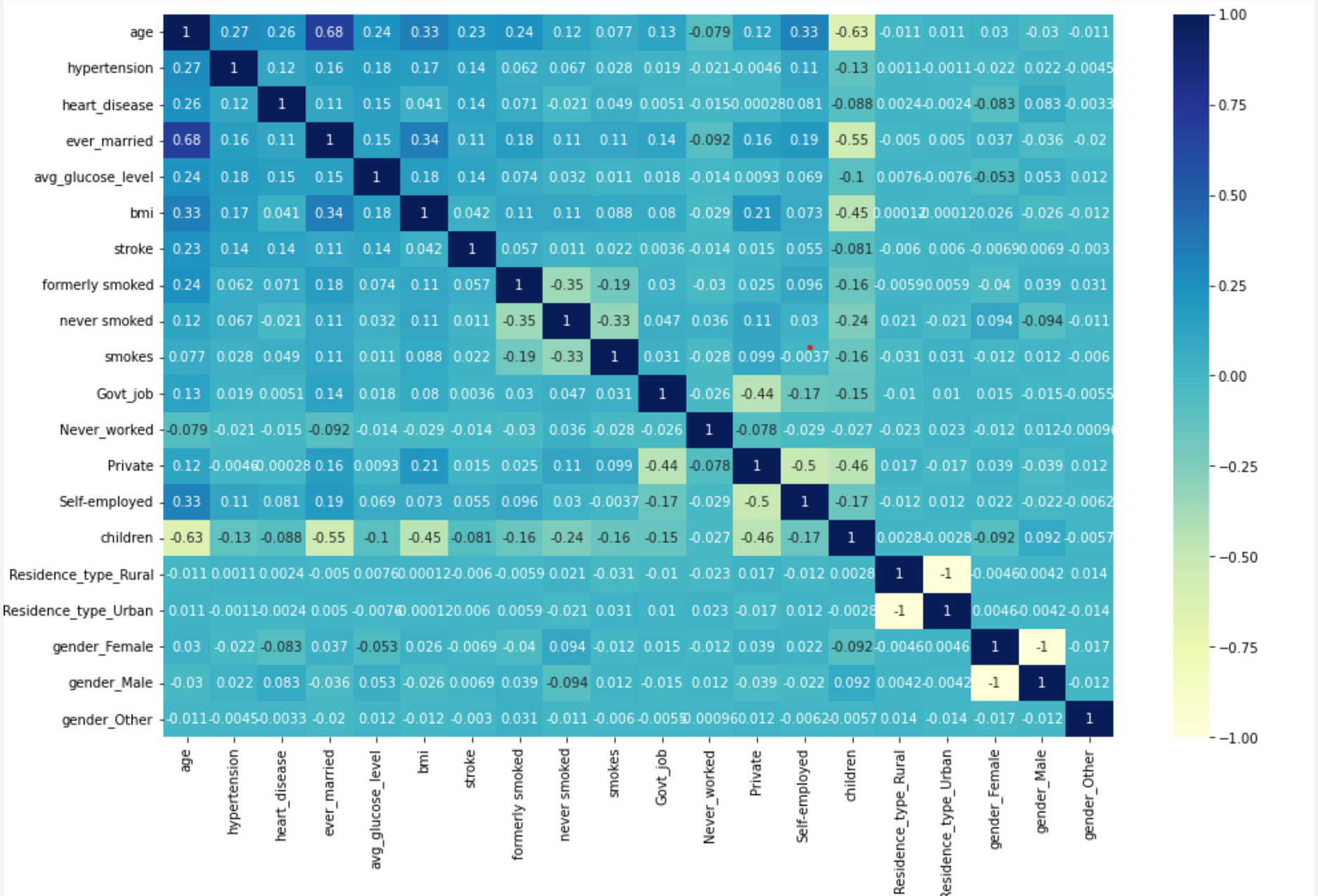




Correlation Matrix



Heatmap of the Data



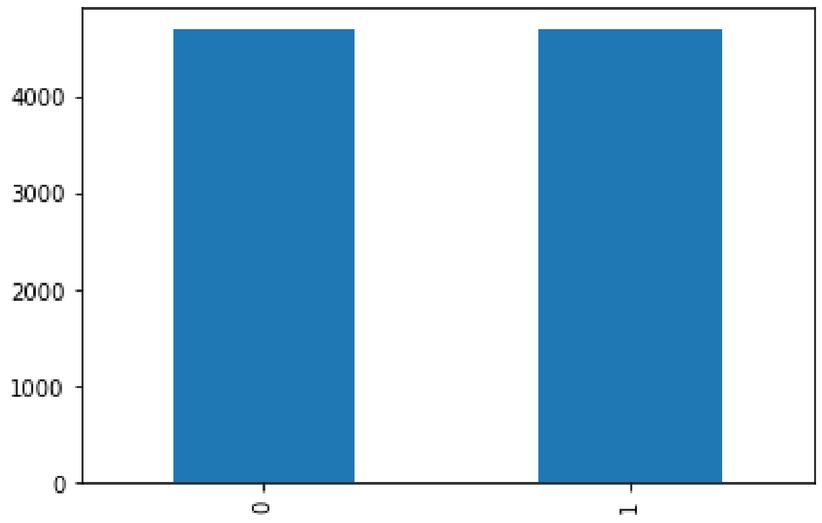
Feature Engineering

	age	hypertension	heart_disease	ever_married	avg_glucose_level	bmi	stroke	formerly smoked	never smoked	smokes	Govt_job	Never_worked	Private	Self-employed
0	67.0	0	1	1	228.69	36.6	1	1	0	0	0	0	1	0
2	80.0	0	1	1	105.92	32.5	1	0	1	0	0	0	1	0
3	49.0	0	0	1	171.23	34.4	1	0	0	1	0	0	1	0
4	79.0	1	0	1	174.12	24.0	1	0	1	0	0	0	0	1
5	81.0	0	0	1	186.21	29.0	1	1	0	0	0	0	1	0

Self-employed	children	Residence_type_Rural	Residence_type_Urban	gender_Female	gender_Male	gender_Other
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	0	1	1	0	0
1	0	1	0	1	0	0
0	0	0	1	0	1	0

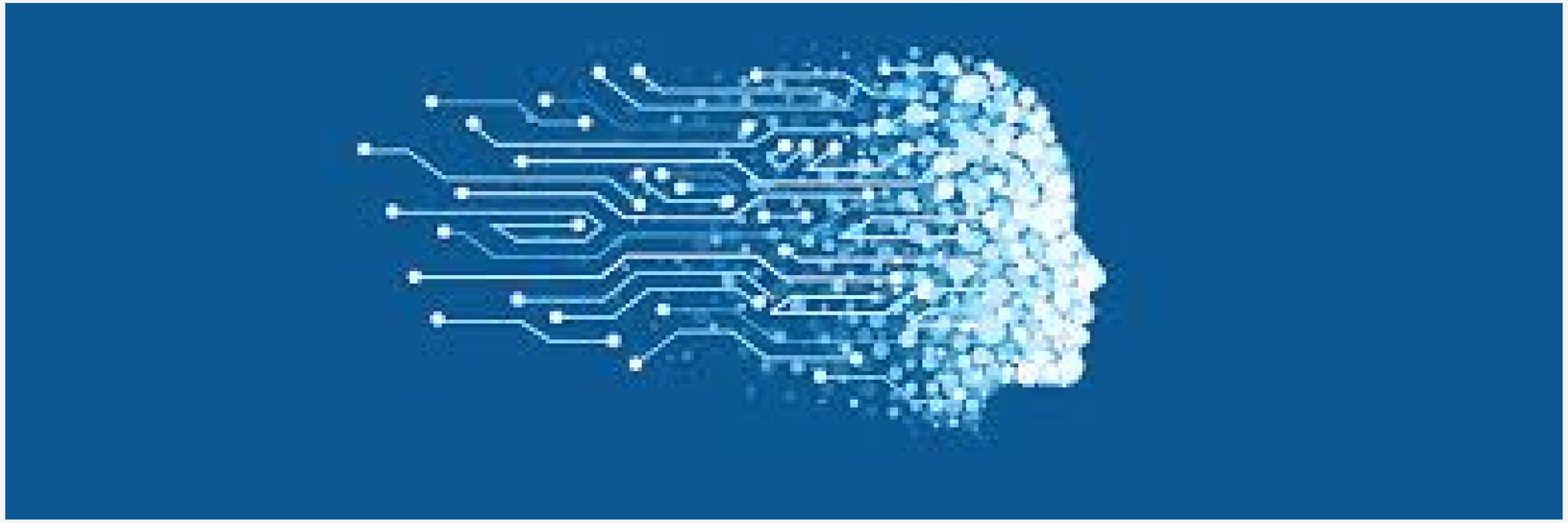
Dummy Variables

```
class 0: (4700, 20)
class 1: (209, 20)
total values of 1 and 0: 0    4700
1    4700
Name: stroke, dtype: int64
```



Over Sampling

Machine Learning Algorithms



- Machine learning can accelerate this process with the help of decision-making algorithms.
- It can categorize the incoming data, recognize patterns and translate the data into insights helpful for better decision making.

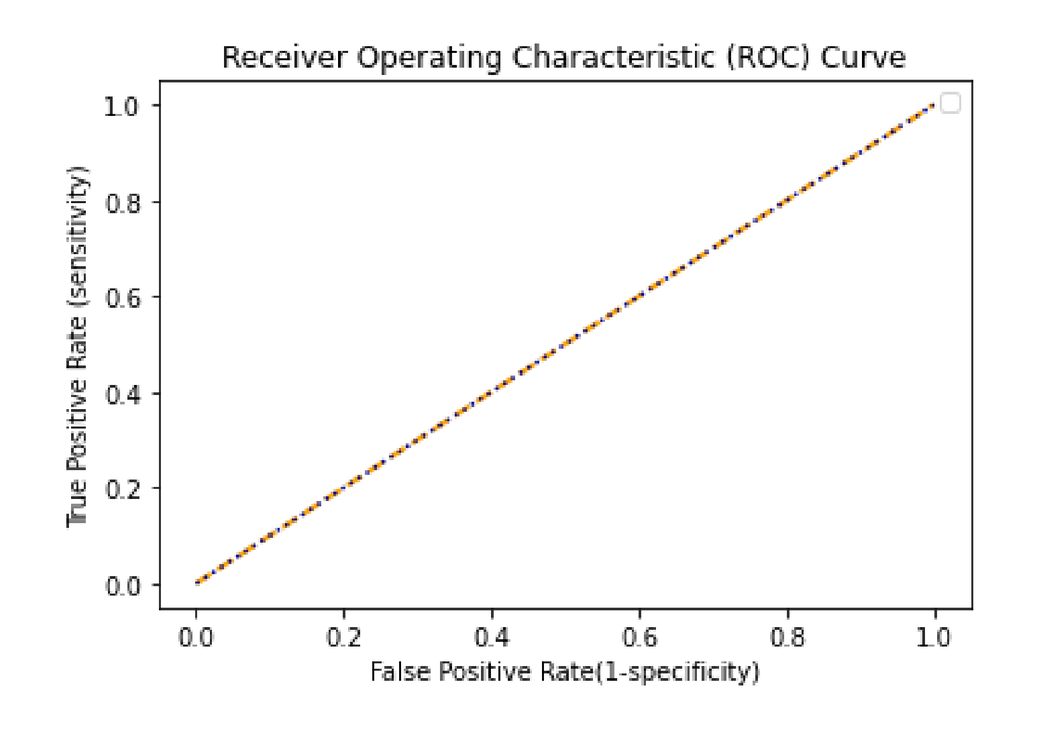
SVM

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

SVM

- Library used : sklearn.svm (SVC)
- Splitting the data into Training & Testing in a ratio of 70:30
- Accuracy achieved: - [0.75496454]
- Precision: [0.69887955]



Random Forest

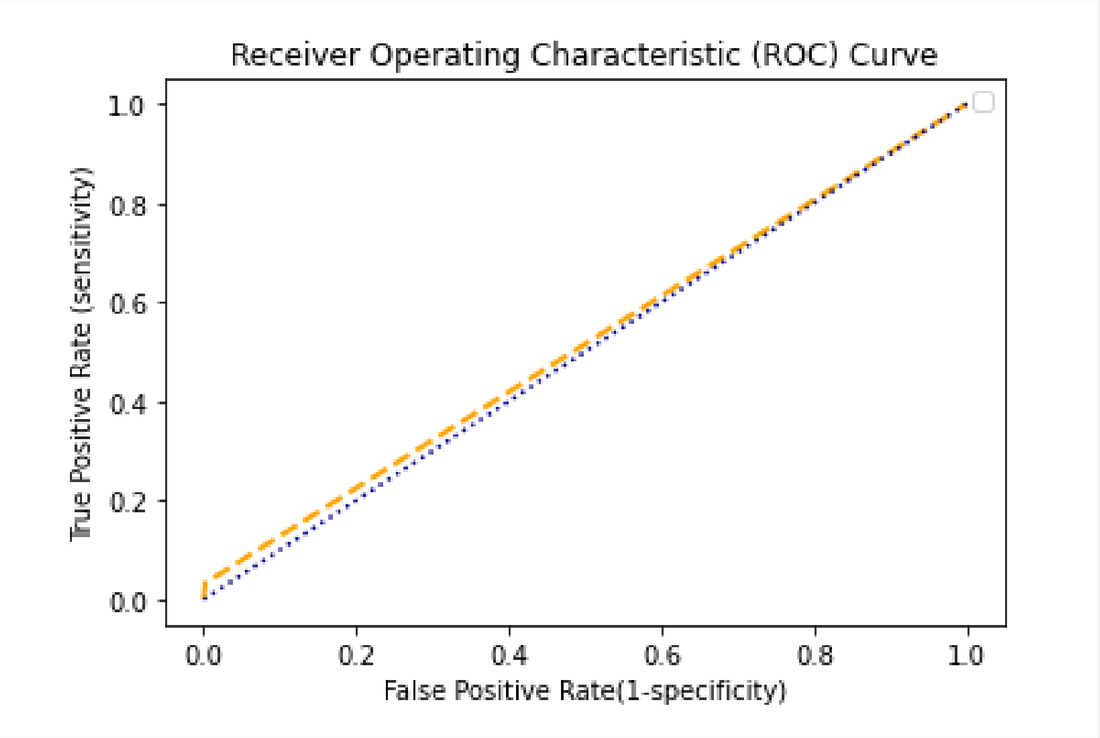
Random Forest belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Random Forest

- Library used : sklearn.ensemble (RandomForestClassifier)
- Splitting the data into Training & Testing in a ratio of 70:30
- Accuracy achieved: - [0.99113475]
- Precision: - [0.982493]



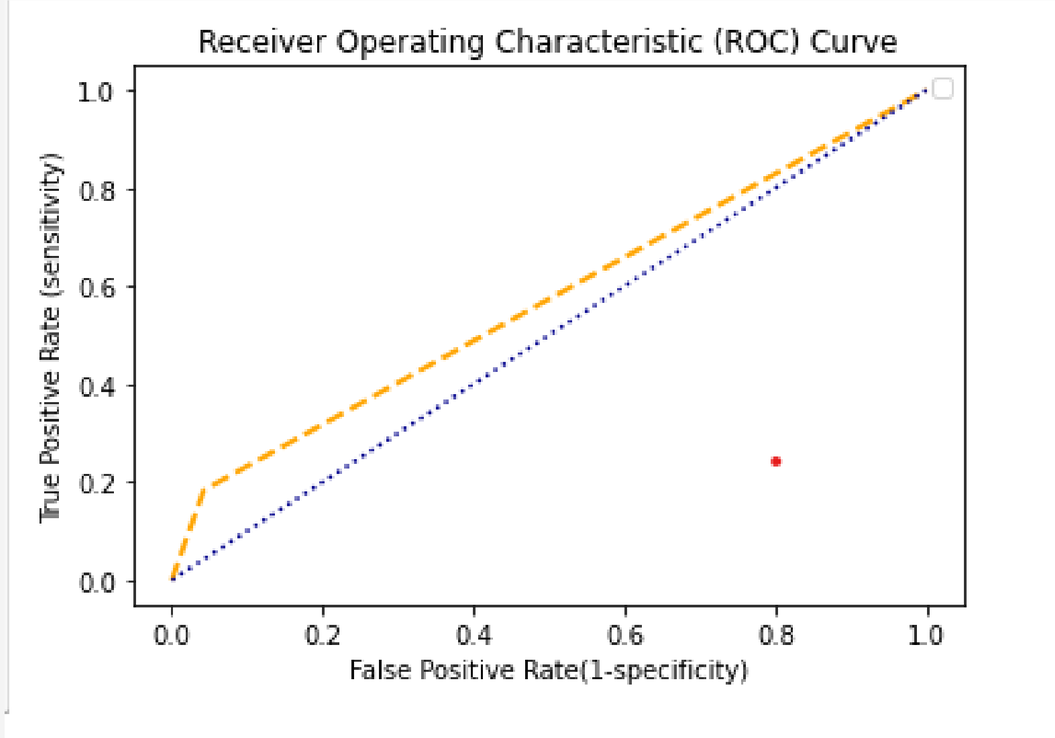
Decision Tree

A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered.
The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization.

The decision tree may not always provide a clear-cut answer or decision. Instead, it may present options so the data scientist can make an informed decision on their own.
Decision trees imitate human thinking, so it’s generally easy for data scientists to understand and interpret the results.

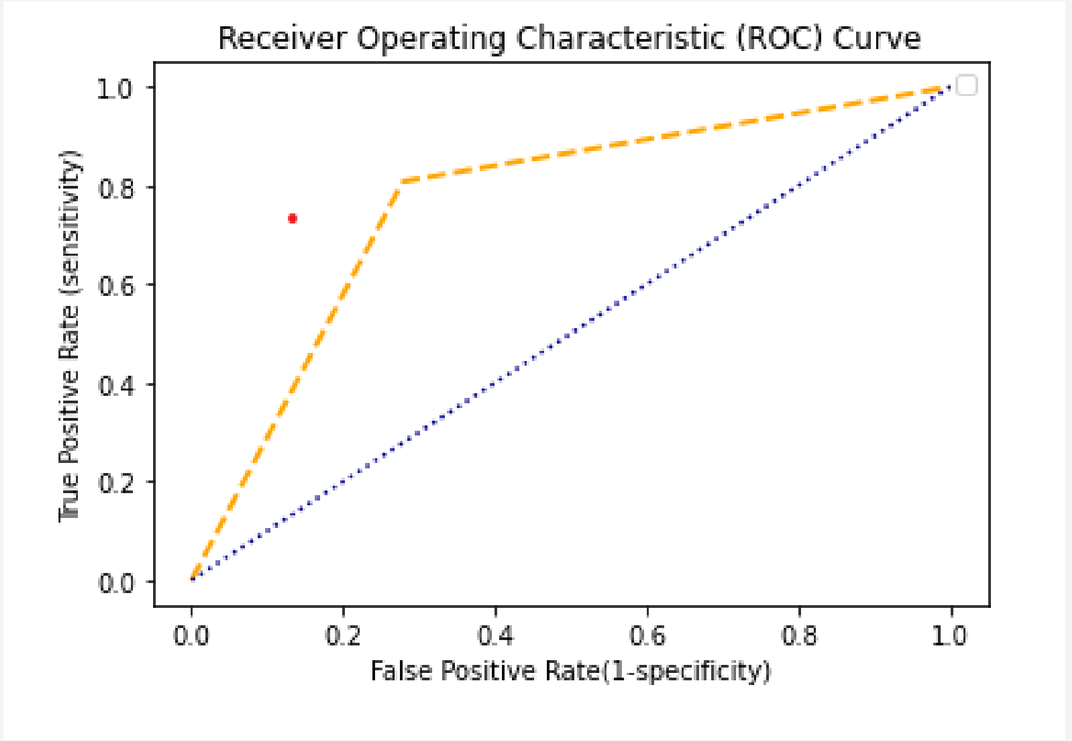
Decision Tree

- Library used : sklearn.tree (DecisionTreeClassifier)
- Splitting the data into Training & Testing in a ratio of 70:30
- Accuracy achieved: - [0.96879433]
- Precision: - [0.93837535]



Logistic Regression

- Library used : sklearn.linear_model (LogisticRegression)
- Splitting the data into Training & Testing in a ratio of 70:30
- Accuracy achieved: - [0.77092199]
- Precision: - [0.72268908]



Logistic Regression

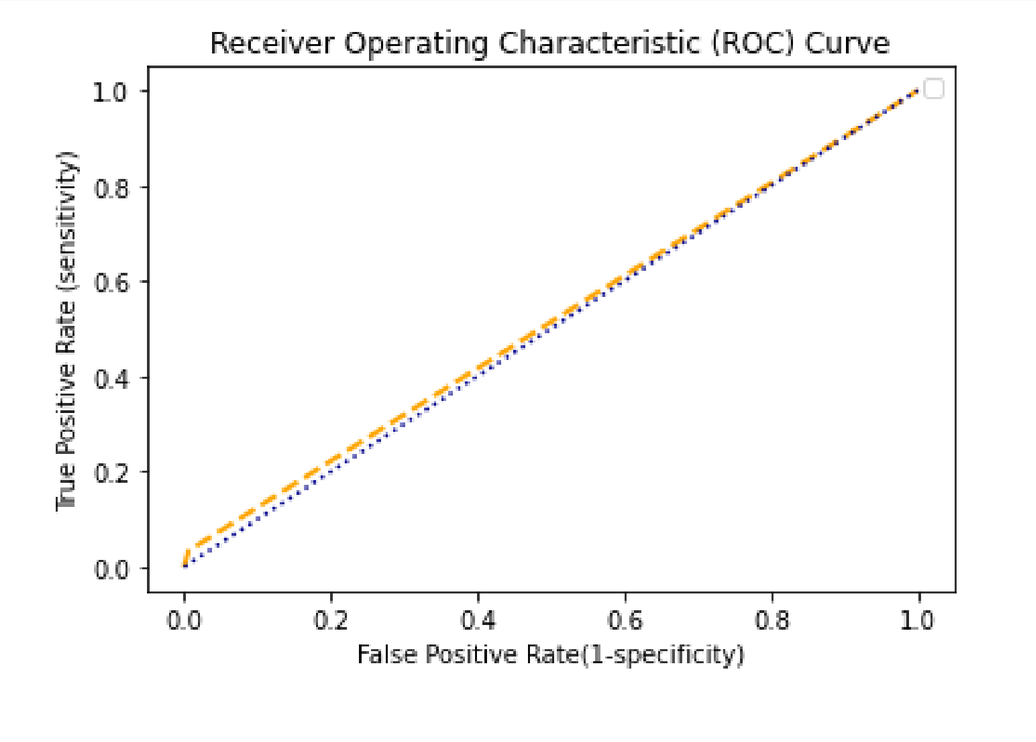
Logistic regression is a data analysis technique that uses mathematics to find the relationships between two data factors. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no.

Below are the some of the benefits of using logistics regression:

- Simplicity
- Speed
- Flexibility
- Visibiity

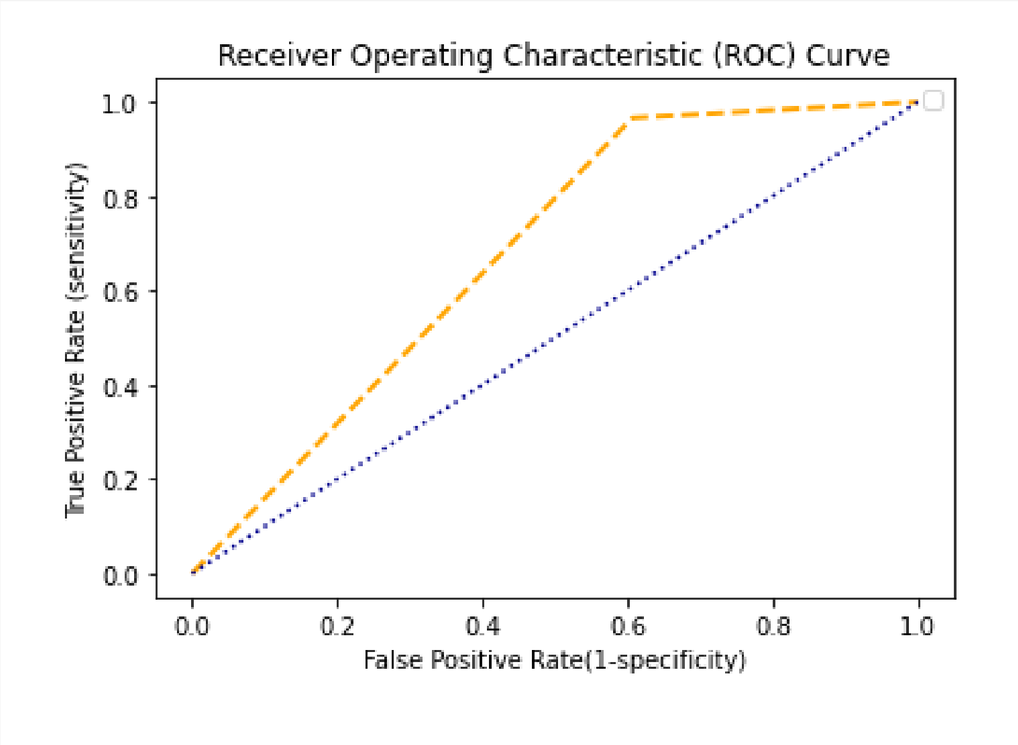
KNN

- Library used : sklearn.neighbors (KNeighborsClassifier)
- Splitting the data into Training & Testing in a ratio of 70:30
- Accuracy achieved: - [0.93049645]
- Precision: - [0.8627451]

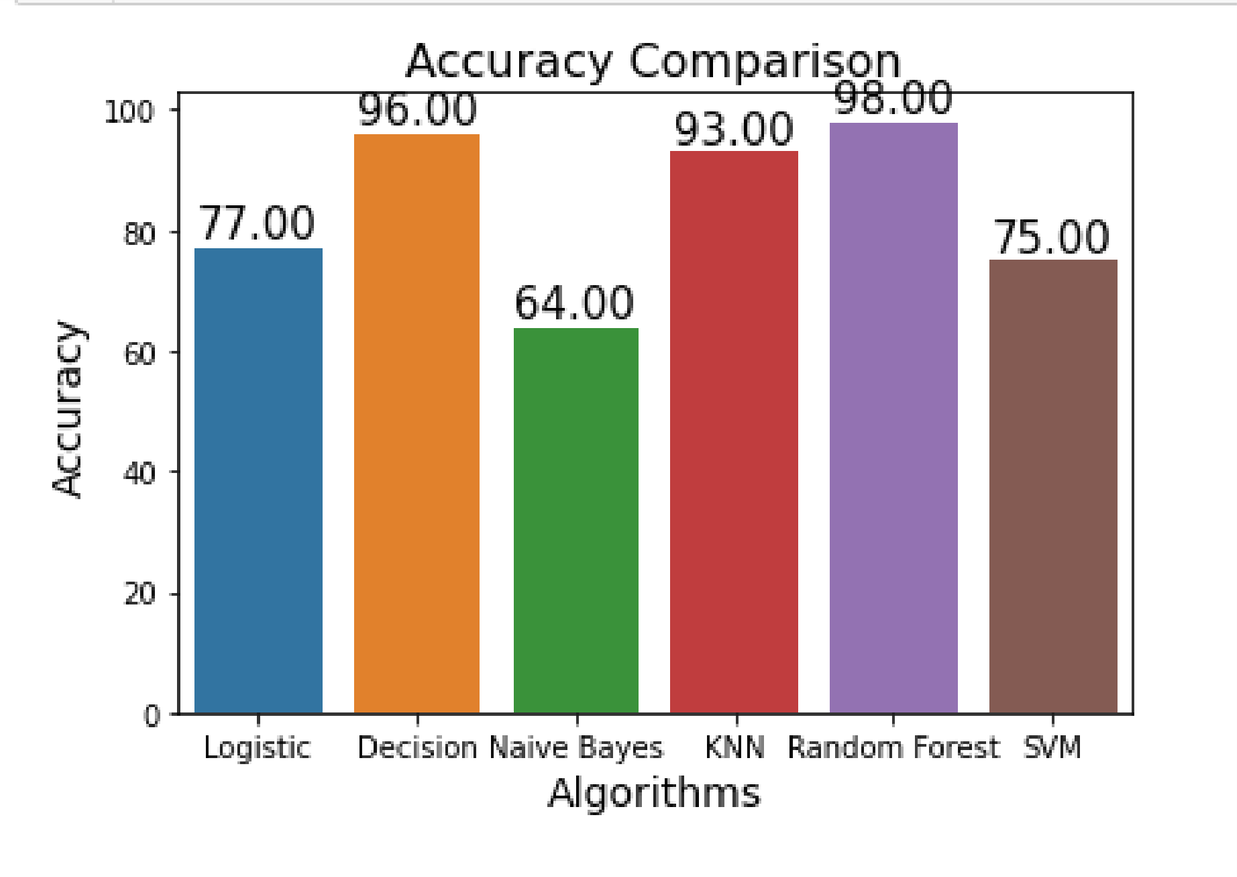


Naive Bayes

- Library used : sklearn.naive_bayes (GaussianNB)
- Splitting the data into Training & Testing in a ratio of 70:30
- Accuracy achieved: - [[0.64326241]
- Precision: - [0.29761905]



Accuracy Comparison



By looking at the above image we can clearly say that the highest accuracy achieved is by Random Forest Classifier with 98% Accuracy and Decision Tree with 96% Accuracy, where as the lowest accuracy achieved by Naive Baye's Classifier of 64%.

Thank You

Feel free to reach out!

Back to Agenda Page

Our Team

- Sanjeev Malik
- Pavan Aditya
- Deepitha
- Sunil Kumar