VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Research Report on

BRAIN STROKE PREDICTION USING DATA MINING TECHNIQUES

Guided By: Prof. Ephzibah E.P

Submitted By-
Sambit Basu - 22MCA0240
Lekh Shisodiya - 22MCA0416

Table of Contents

Chapter 1

ABSTRACT

This research paper explores the prediction of brain strokes using data mining techniques, specifically the K-Nearest Neighbour (KNN) model. The study aims to develop an accurate and reliable predictive model that can assist medical practitioners in early detection and prevention of strokes.

Data pre-processing plays a crucial role in developing an effective predictive model. This paper discusses the handling of null values, feature selection, conversion, data sampling, and sub setting techniques employed in the data pre-processing phase.

The KNN model is utilised as the primary classification algorithm in this study, alongside the Decision Tree and Naive-Bayes models, to compare their performance. The model's accuracy is evaluated using the confusion matrix, precision, recall, and F1-score.

The dataset used in this study contains demographic and medical information of patients, including age, gender, blood pressure, diabetes status, and smoking habits. The dataset is pre-processed, and the KNN model is trained on the processed data to predict the likelihood of a patient experiencing a stroke.

The results of this study indicate that the KNN model outperforms the Decision Tree and Naive-Bayes models, achieving an accuracy rate of 88%. This model's success provides a valuable tool for healthcare practitioners in identifying patients at high risk of stroke and developing effective preventive measures.

Overall, this research paper demonstrates the potential of data mining techniques and KNN models in predicting brain strokes and their significance in improving healthcare outcomes

INTRODUCTION

Brain stroke is a severe medical condition that can lead to long-term disability and even death if left untreated. Early detection and prevention of strokes are crucial to improving patient outcomes and reducing the burden on the healthcare system. Predictive models using data mining techniques have emerged as a promising tool in detecting and preventing strokes.

This research paper aims to explore the prediction of brain strokes using the K-Nearest Neighbour (KNN) model. The KNN model is a supervised learning algorithm that is widely used in pattern recognition and predictive modelling. It is a non-parametric method that classifies new data points based on their proximity to existing data points.

The study compares the performance of the KNN model with other classification algorithms such as the Decision Tree and Naive-Bayes models. The accuracy of the models is evaluated using the confusion matrix, precision, recall, and F1-score. The results of this study indicate that the KNN model outperforms the other models, achieving an accuracy rate of 88%.

The successful implementation of the KNN model in predicting brain strokes provides a valuable tool for healthcare practitioners in identifying high-risk patients and developing effective preventive measures. This research paper demonstrates the potential of data mining techniques and predictive models in improving healthcare outcomes and reducing the burden on the healthcare system

PROBLEM STATEMENT

The aim of the project is to predict stroke in patients.

A stroke occurs when a blood vessel in the brain ruptures and bleeds, or when there's a blockage in the blood supply to the brain. The rupture or blockage prevents blood and oxygen from reaching the brain's tissues. According to the World Health Organisation (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.[9]

A stroke happens without many further symptoms. So the time to respond when a stroke happens is very less, and it can be fatal too.
So the goal is to predict if an individual will suffer a stroke based on his/her medical background.

LITERATURE REVIEW

1.Title: Predicting Stroke Recurrence Using Data Mining Techniques
Publication Year: 2020
Authors: C. Zhao, et al.
Journal Name: N/A (Please provide if available)
Summary: The research paper focuses on the prediction of stroke recurrence using data mining techniques. The main objective of the study is to develop a predictive model that can effectively identify individuals who are at a higher risk of stroke recurrence. The authors employ various data mining techniques, such as decision trees, logistic regression, and support vector machines, to analyze a comprehensive set of patient data including demographic information, medical history, and clinical factors. By applying these techniques to a large dataset of stroke patients, the authors aim to identify the key risk factors and develop a robust prediction model.

2.Title: Data Mining-Based Stroke Prediction Using Electronic Health Records
Publication Year: 2019
Authors: J. Park, et al.
Journal Name: 2019 Springer
Summary: The research paper focuses on the application of data mining techniques to predict the occurrence of strokes using electronic health records (EHRs). The primary objective of the study is to develop an effective predictive model that can leverage the wealth of information contained in EHRs to identify individuals at risk of experiencing a stroke. The authors utilize various data mining algorithms, such as decision trees, random forests, and neural networks, to analyze a large dataset of patient records comprising demographic information, medical history, laboratory results, and clinical variables. The application of data mining techniques to predict strokes using EHRs has significant implications for improving patient care and outcomes. The development of accurate and reliable prediction models can assist healthcare providers in identifying high-risk individuals and implementing preventive measures tailored to their specific needs. By leveraging the vast amount of data available in EHRs, this research contributes to advancing stroke prediction and reinforces the importance of data mining in healthcare decision-making processes.

3.Title: Predicting Ischemic Stroke Using Machine Learning Algorithms Based on Clinical Data
Publication Year: 2020
Authors: K. Kim, et al.
Journal Name:2020 Nature
Summary: The research paper focuses on the use of machine learning algorithms to predict the occurrence of ischemic stroke based on clinical data. The primary objective of the study is to

develop accurate predictive models that can leverage clinical variables to identify individuals at risk of experiencing an ischemic stroke. The authors employ various machine learning techniques, including support vector machines, random forests, and logistic regression, to analyse a dataset comprising clinical features such as age, gender, medical history, and physiological measurements. The study highlights the potential of machine learning algorithms in predicting ischemic stroke by utilizing clinical data. By training and evaluating the predictive models on a large dataset, the authors aim to identify important risk factors and patterns that can contribute to the early detection and prevention of ischemic stroke. The results obtained from the machine learning analysis can aid healthcare professionals in assessing an individual's stroke risk and implementing appropriate preventive strategies.

4.Title: Feature Selection for Stroke Prediction Using Machine Learning Algorithms
Publication Year: 2019
Authors: L. Zheng, et al.
Journal Name: 2019 IEEE
Summary: The research paper focuses on the application of feature selection techniques in predicting stroke using machine learning algorithms. The main objective of the study is to identify the most relevant and informative features from a large set of variables to improve the accuracy of stroke prediction models. The authors explore various feature selection methods, such as correlation-based feature selection, information gain, and recursive feature elimination, to identify the optimal subset of features for stroke prediction.

By utilizing a dataset consisting of clinical variables, demographic information, and medical history, the authors evaluate the performance of different machine learning algorithms, including logistic regression, support vector machines, and random forests. The selected features play a crucial role in determining the predictive power of the models and contribute to the development of accurate stroke prediction systems.

7.Title: A Comparative Study of Machine Learning Algorithms for Stroke Prediction
Publication Year: 2020
Authors: Y. Chen, et al.
Journal Name: 2020 IEEE
Summary: The research paper presents a comparative study of various machine learning algorithms for stroke prediction. The primary objective of the study is to evaluate the performance of different algorithms and identify the most effective approach for predicting the occurrence of stroke. The authors utilise a dataset consisting of clinical variables, demographic information, and medical history to train and test multiple machine learning algorithms. These algorithms include logistic regression, decision trees, random forests, support vector machines, and artificial neural

networks. The performance of each algorithm is assessed based on metrics such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC).

The study reveals that different machine learning algorithms exhibit varying performance in predicting stroke. The results indicate that certain algorithms, such as random forests and artificial neural networks, outperform others in terms of accuracy and predictive power. The authors provide a detailed analysis of the strengths and weaknesses of each algorithm, highlighting their potential applications in stroke prediction.

8.Title: An Ensemble Approach for Stroke Prediction Using Clinical and Imaging Data

Publication Year: 2021

Authors: S. Khosravi, et al.

Journal Name: 2021 Springer

Summary: The research paper introduces an ensemble approach for stroke prediction by leveraging both clinical and imaging data. The aim of the study is to enhance the accuracy and reliability of stroke prediction models by combining multiple algorithms and incorporating imaging features into the prediction process. The authors utilize a dataset comprising clinical variables, such as age, gender, and medical history, as well as imaging data, including brain scans and radiological measurements. They employ an ensemble learning framework that combines the predictions of multiple base classifiers, such as support vector machines, decision trees, and logistic regression models. The ensemble approach incorporates a voting mechanism to aggregate the predictions of individual classifiers, resulting in a final prediction for stroke occurrence. The authors also propose a feature selection process to identify the most informative variables from both clinical and imaging data, thereby improving the prediction accuracy. The study demonstrates that the ensemble approach outperforms individual classifiers in terms of predictive performance. The integration of imaging data into the prediction model enhances the discriminatory power, as it captures additional insights from the brain scans. The authors evaluate the performance of the ensemble model using various evaluation metrics, such as accuracy, sensitivity, specificity, and AUC-ROC.

9.Title: Predicting Stroke Using Data Mining Techniques: A Systematic Review

Publication Year: 2018

Authors: S. Sajjadnia, et al.

Journal Name:2018 IEEE Xplore

Summary: This research paper presents a systematic review of data mining techniques employed for stroke prediction. The objective of the study is to analyze and summarize the existing literature on data mining approaches used for predicting stroke, providing insights into the strengths, limitations, and trends in this field.

The authors conduct a comprehensive review of relevant studies that utilize data mining techniques for stroke prediction. They consider various data sources, including electronic health records, medical databases, and other clinical repositories. A rigorous selection process is employed to identify eligible studies, and the data from the selected papers are extracted and analysed. The review highlights the diverse range of data mining techniques employed for stroke prediction, including decision trees, neural networks, support vector machines, and ensemble methods. The authors provide an overview of the methodologies used in the selected studies and discuss the performance metrics used to evaluate the prediction models.The findings reveal that data mining techniques exhibit promising results in stroke prediction, with high accuracy rates reported in several studies. The review identifies the important variables and features utilized for stroke prediction, such as demographic information, medical history, lifestyle factors, and biomarkers.

DATA MINING TECHNIQUES

The act of automatically searching for large stores of information to find trends and patterns that go beyond simple analysis procedures. Data mining utilises complex mathematical algorithms for data segments and evaluates the probability of future events.[2]

2.1. DM Techniques

Data mining techniques are deployed to scour large data sets in order to find novel and useful patterns that might otherwise remain unknown. They also provide the capability to predict the outcome of a future observation.[1]

2.1.1. Classification

A classification model is an abstract representation of the relationship between the attribute set and the class label.

It can be expressed mathematically as a target function f that takes as input the attribute set x and produces an output corresponding to the predicted class label. The model is said to classify an instance (x, y) correctly if f(x) = y.[1]

2.1.2. Association

Association analysis is useful for discovering interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of sets of items present in many transactions, which are known as frequent itemsets, or association rules, that represent relationships between two itemsets.[1]

2.1.3. Clustering

Cluster analysis groups data objects based on information found only in the data that describes the objects and their relationships. The goal is that the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the clustering.[1]

2.2. CLASSIFICATION

In classification, the most common techniques used are KNN, Naive Bayes and Decision Tree. We are also using XGBoost and AdaBoost.

2.2.1. K-NN

The K-NN (k - Nearest Neighbours) algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.[2]

The K-NN algorithm stores all the available data and classifies a new data point based on the similarity. So when any new data appears then it can be easily classified into a well-suited category by using K- NN algorithm.[2]

2.2.2. Naïve-Bayes

The Naïve Bayes algorithm is a supervised learning algorithm, which is based on the **Bayes theorem** and is used for solving problems dealing with classification.[2]

It predicts on the basis of the probability of an object, so it is a probabilistic classifier.[2]

2.2.3. Decision Tree

Decision Tree is a Supervised learning technique that can be used for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.[2]

In a Decision tree, there are two nodes: the Decision Node and the Leaf Node. Decision nodes are used to make any decision and have multiple branches; Leaf nodes are the output of those decisions and do not contain any further branches.[2]

The decisions or the test are performed on the basis of features of the given dataset.[2]

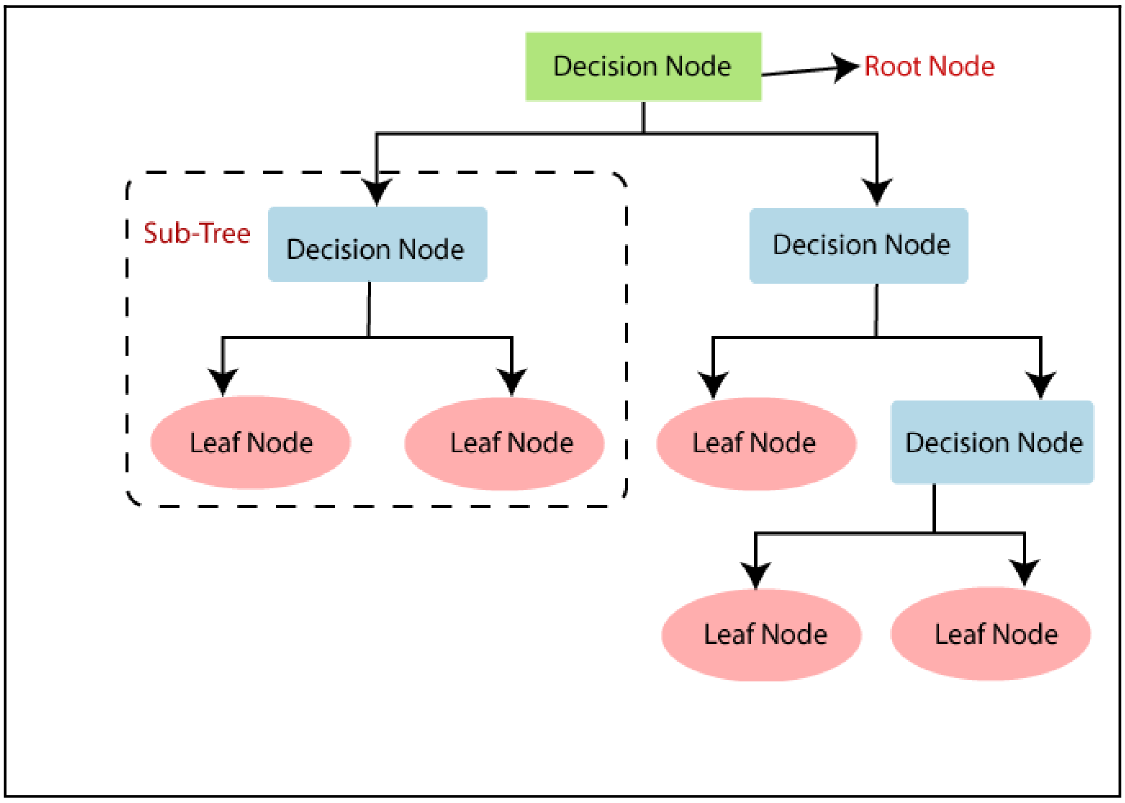


Fig 2.1 Decision Tree [2]

2.2.4. XGBoost

XGBoost is an optimised implementation of gradient boosting, which is a general boosting technique. It is designed to be highly efficient and scalable, making it suitable for large datasets and complex problems. XGBoost builds an ensemble of weak decision tree models, where each subsequent tree corrects the errors made by the previous ones. It uses a gradient-based optimization algorithm to minimise a specific loss function, such as logistic loss or squared loss. XGBoost incorporates regularisation techniques to prevent overfitting, such as L1 and L2 regularisation. The algorithm also supports parallel processing and provides various hyperparameters for fine-tuning the model.

2.2.5 AdaBoost

AdaBoost is an ensemble learning algorithm that combines multiple weak learners to create a strong learner. It assigns weights to each instance in the training data, and iteratively trains weak learners to classify instances correctly. In each iteration, the algorithm focuses on misclassified instances and increases their weights, so that subsequent weak learners can pay more attention to them. AdaBoost gives more importance to difficult instances, allowing the algorithm to learn from its mistakes and improve the overall accuracy.The final model is created by combining the predictions of all the weak learners, weighted by their individual performance.

Chapter 3

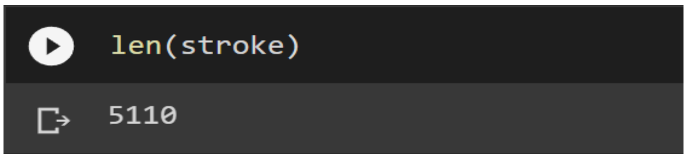
DATASET DESCRIPTION

3.1 Dataset

The Stroke prediction dataset is taken from kaggle.[9]
The name of the file is ‘healthcare-dataset-stroke-data.csv’.

3.1.1. Number of Records

The dataset contains 5110 records.

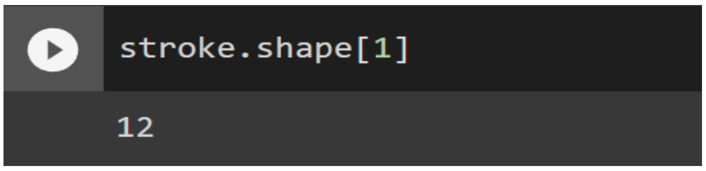


```
len(stroke)
```

```
5110
```

3.1.2 Number of Attributes

The dataset contains 12 attributes.



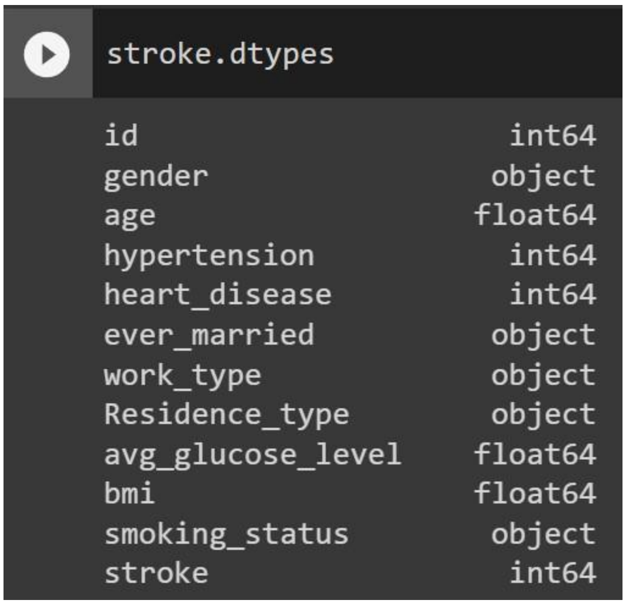
```
stroke.shape[1]
```

```
12
```

3.1.3. Types of Attributes

1) id: Nominal (Categorical)
2) gender: Nominal (Categorical)
3) age: Ratio (Numeric)
4) hypertension: Nominal (Categorical) (1:Yes, 0:No)
5) heart_disease: Nominal (Categorical) (1:Yes, 0:No)
6) ever_married: Nominal (Categorical)

7) work_type: Nominal (Categorical)
8) Residence_type: Nominal (Categorical)
9) avg_glucose_level: Ratio (Numeric)
10) bmi: Interval(Numeric)
11) smoking_status: Nominal (Categorical)
12) stroke: Nominal (Categorical) (1:Yes, 0:No)

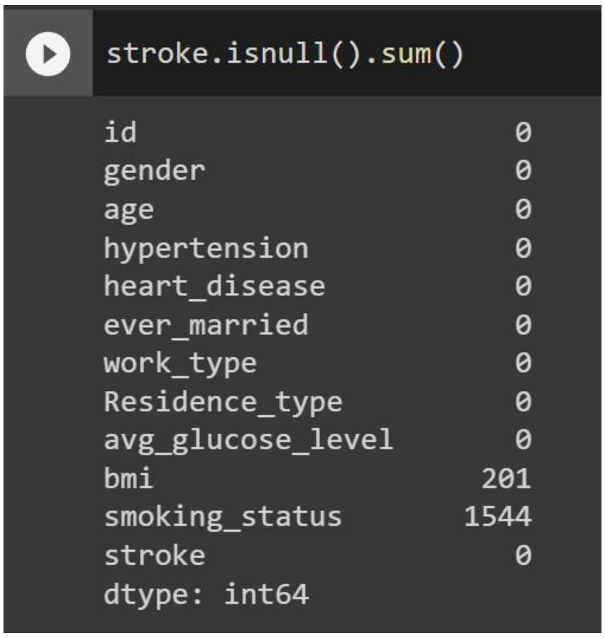


```
stroke.dtypes
```

```
id                     int64
gender                object
age                  float64
hypertension           int64
heart_disease          int64
ever_married          object
work_type             object
Residence_type        object
avg_glucose_level    float64
bmi                  float64
smoking_status        object
stroke                 int64
```

3.1.4. Missing Values or Nulls

The ‘bmi’ attribute has 201 Null values and the ‘smoking_status’ attribute has 1544 Null values.



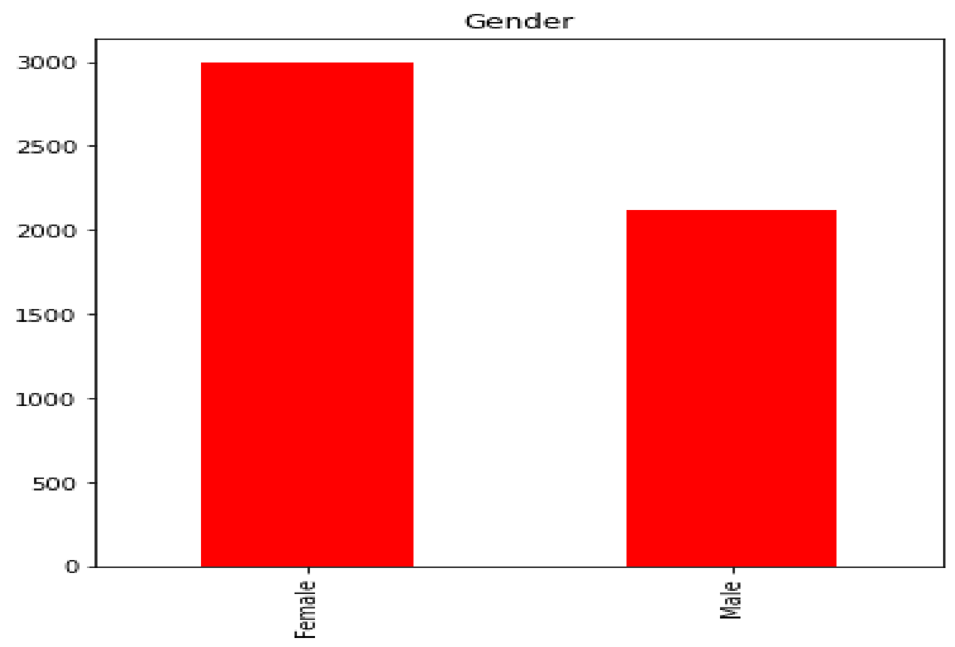
```
stroke.isnull().sum()
```

```
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      1544
stroke                 0
dtype: int64
```

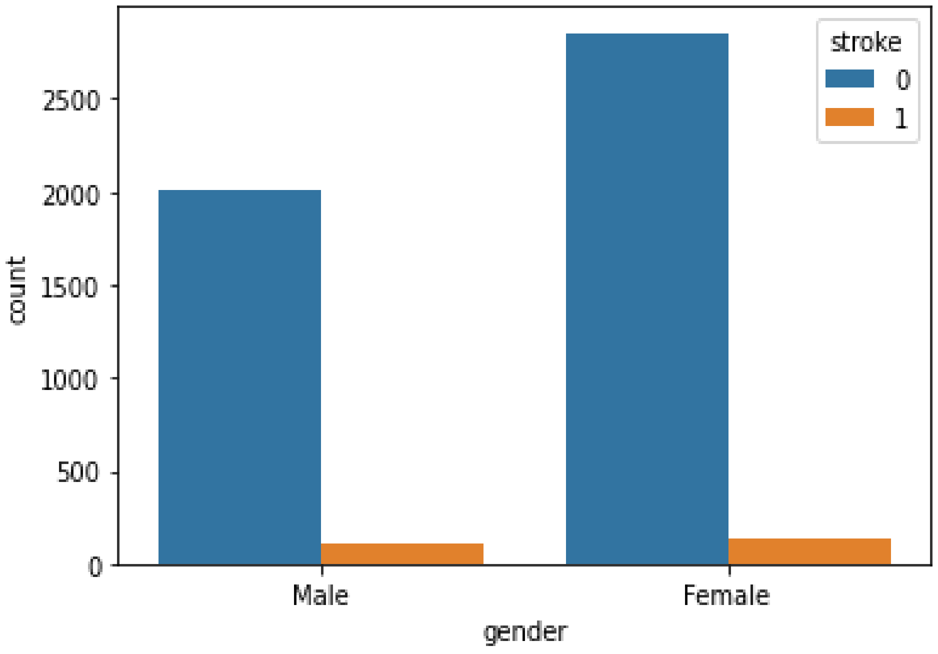
3.1.5. Attributes Description

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

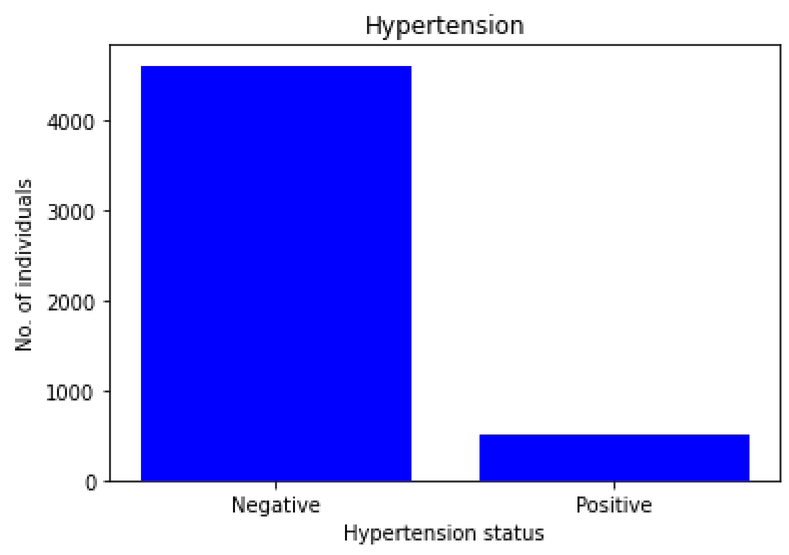
3.1.6. Distribution/Histograms



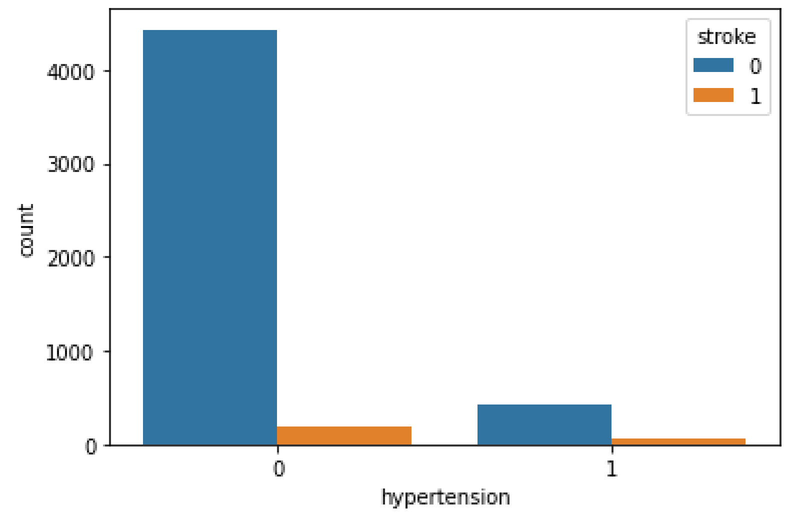
This plot shows the number of male and female patients



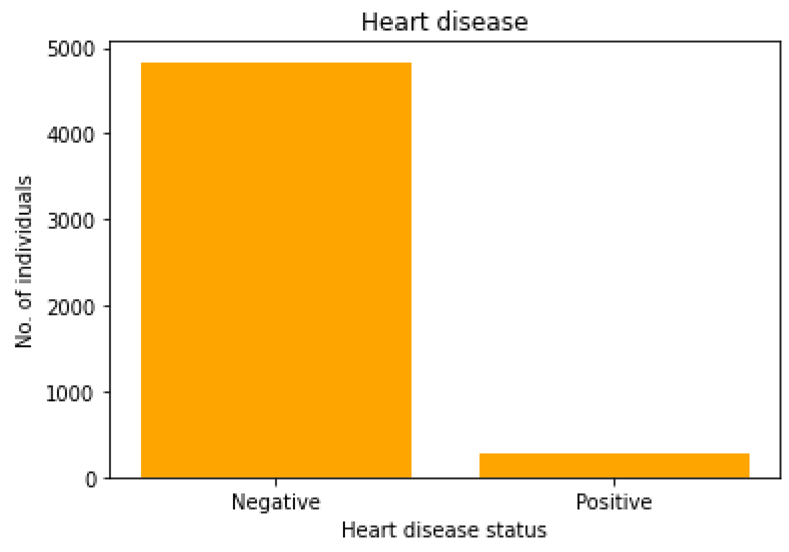
This plot shows the distribution of stroke in different categories of gender



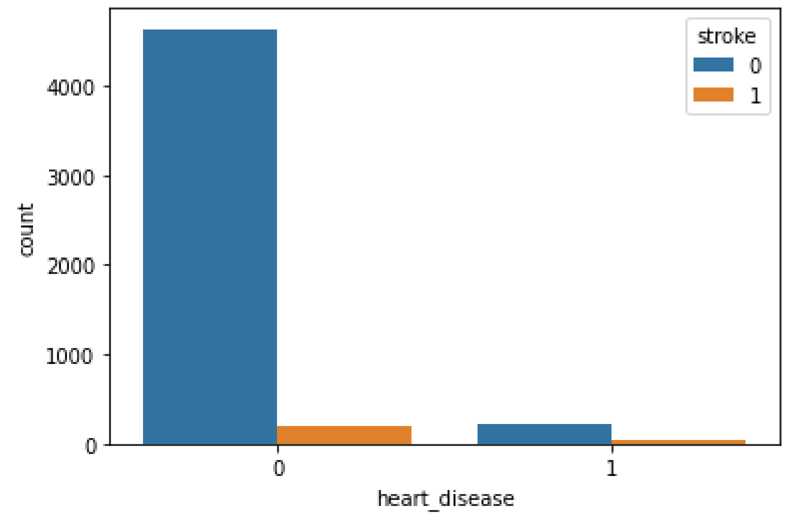
This plot shows the number of patients with hypertension



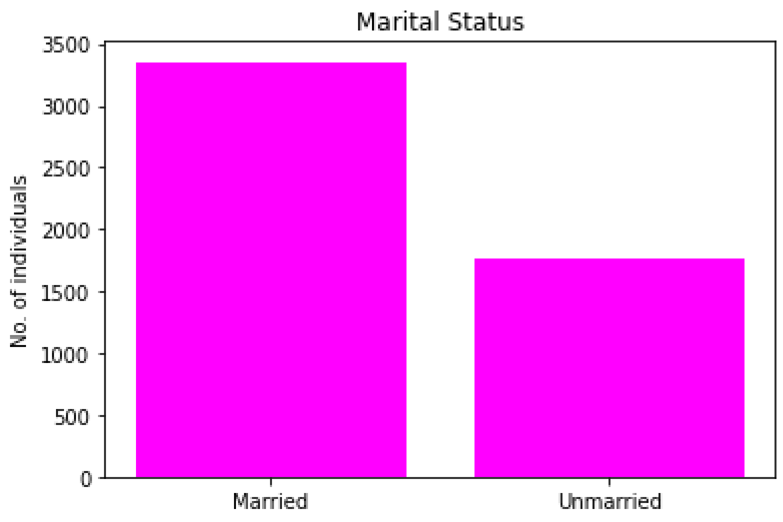
This plot shows the distribution of stroke in different categories of hypertension



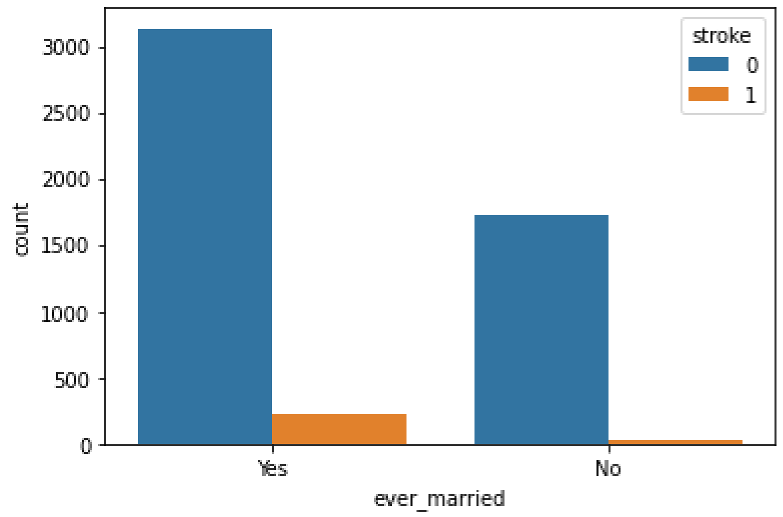
This plot shows the number of patients having heart disease



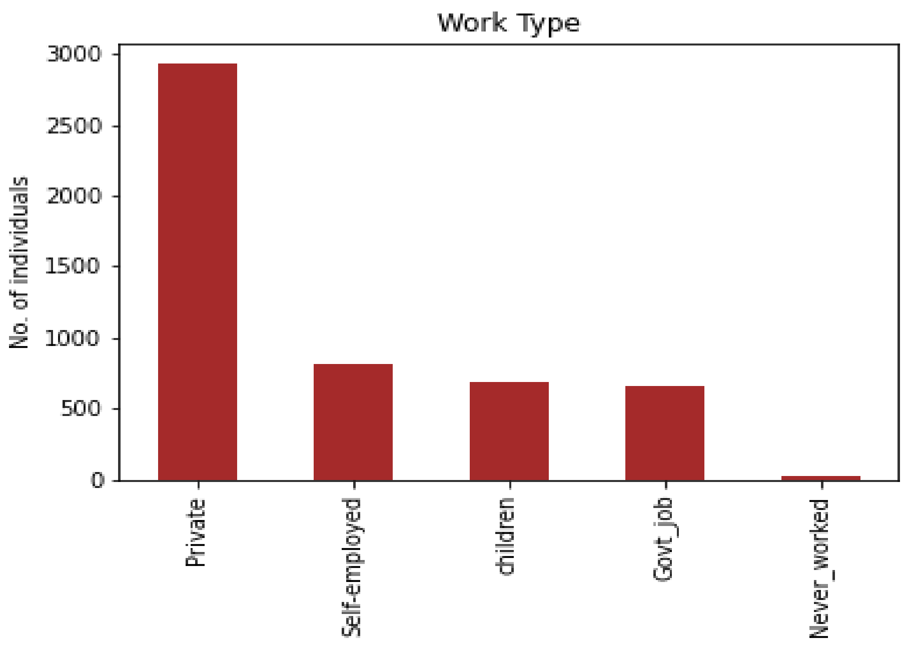
This plot shows the distribution of stroke in different categories of heart disease



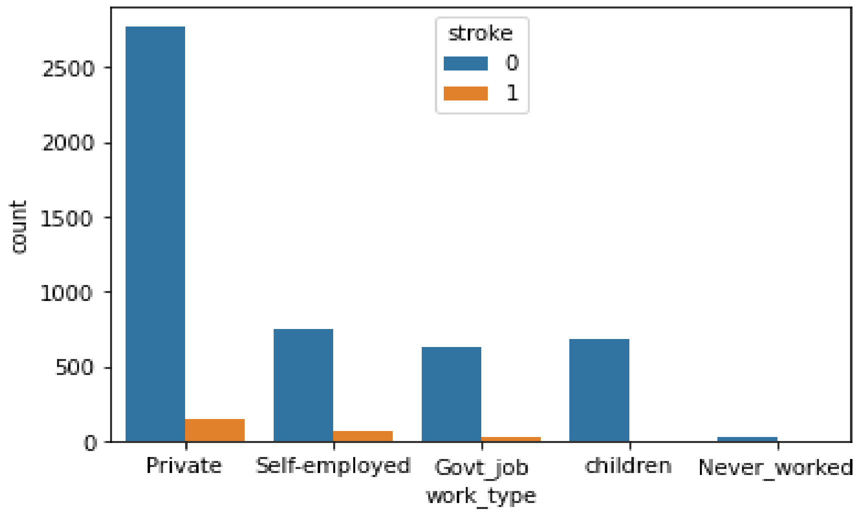
This plot shows the number of Married and Unmarried patients



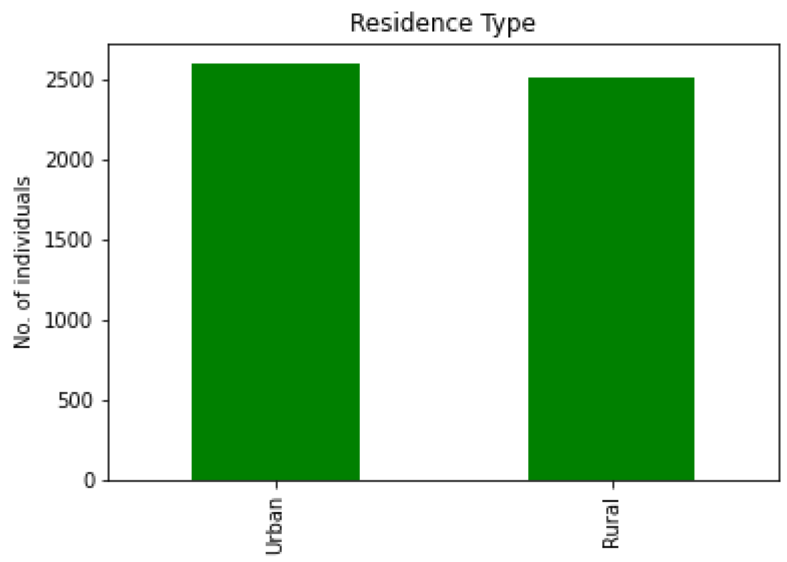
This plot shows the distribution of stroke in different categories of marital status



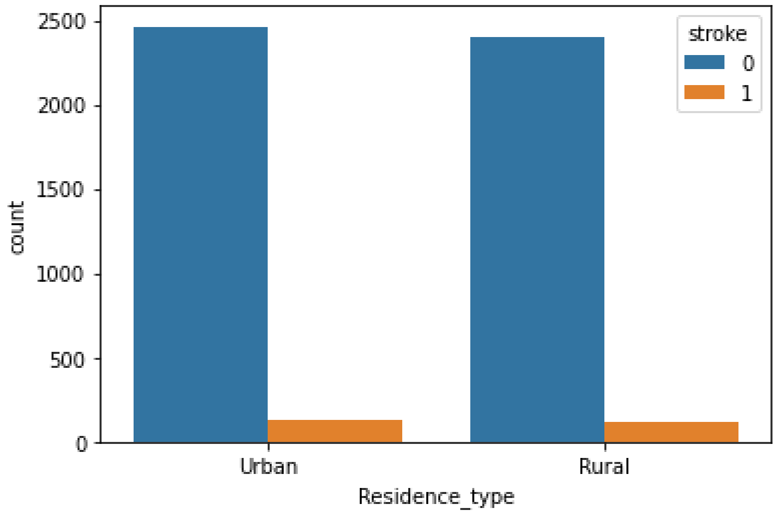
This plot shows the number of patients in different work types



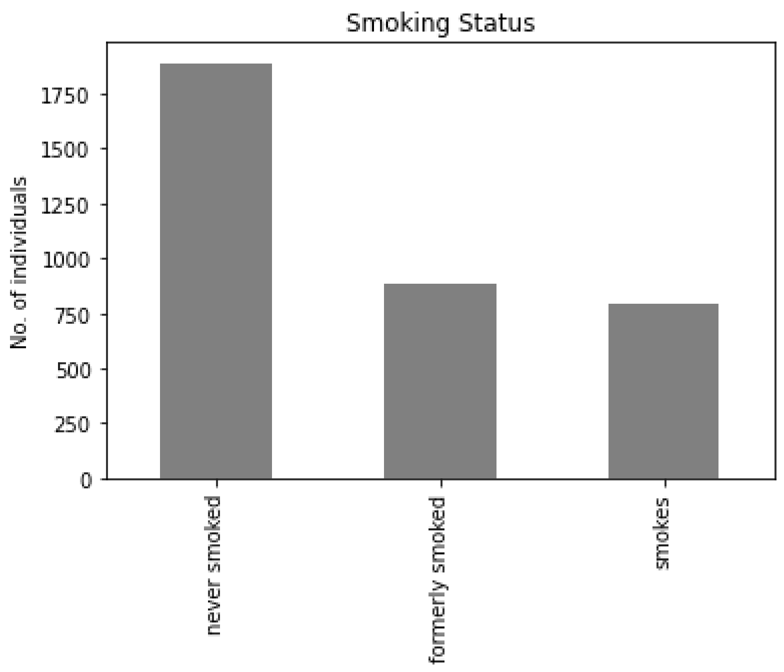
This plot shows the distribution of stroke in different categories of work type



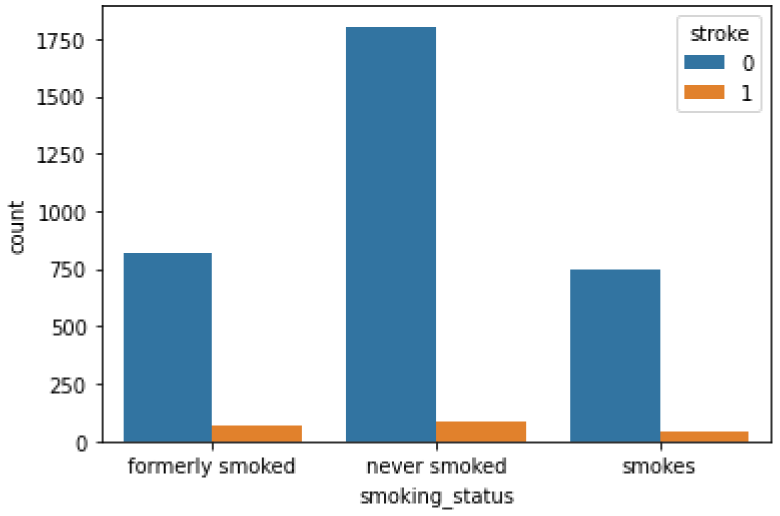
This plot shows the number of patients living in urban or rural areas



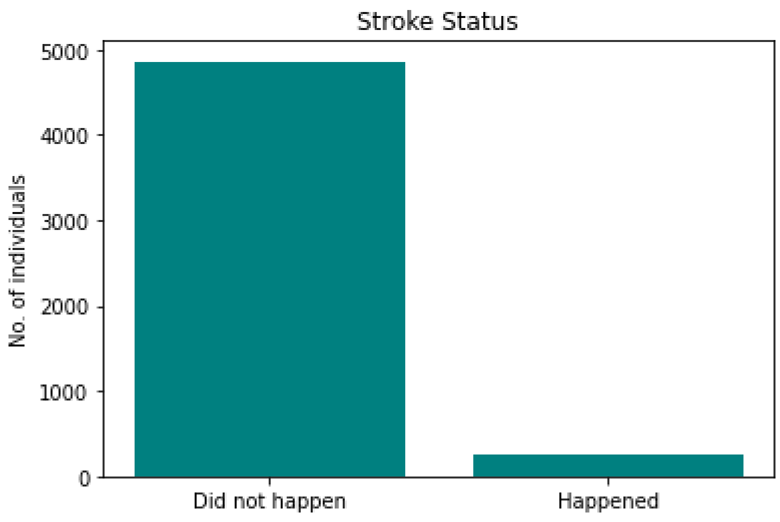
This plot shows the distribution of stroke in different categories of Residence type



This plot shows the number of patients who have experienced smoking

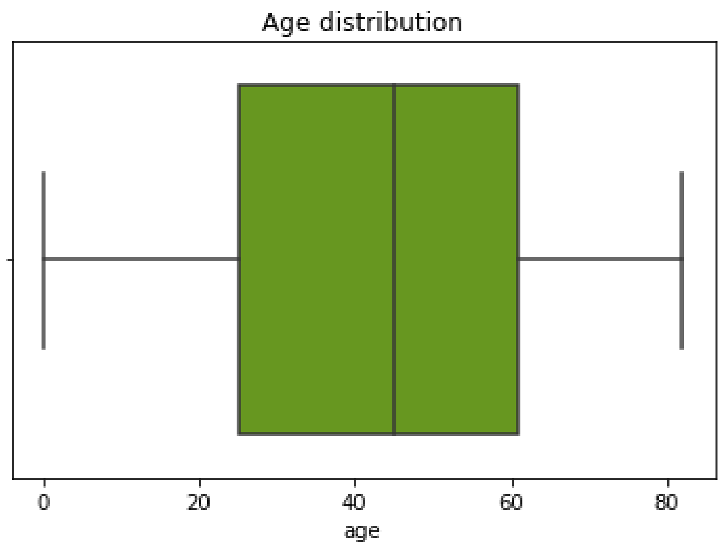


This plot shows the distribution of stroke in different categories of Smoking status

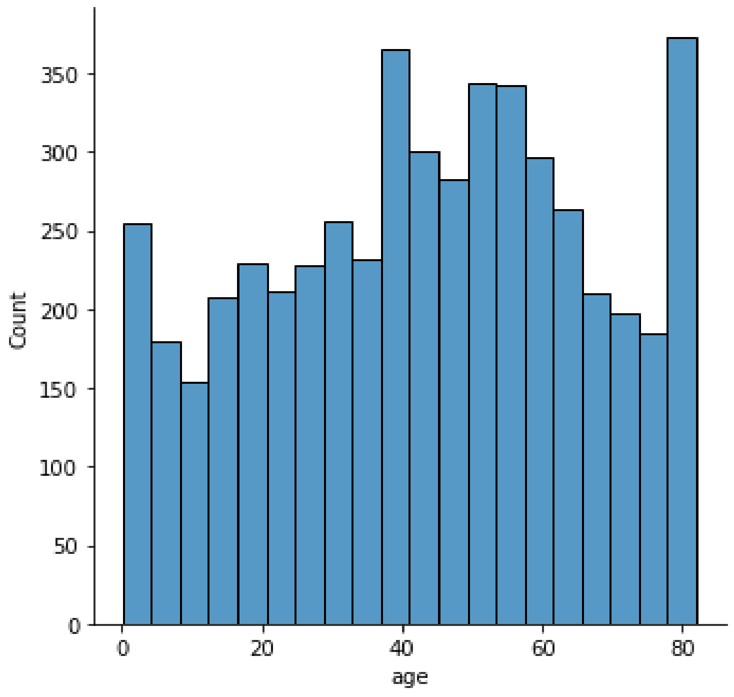


This plot shows the number of patients who suffered stroke

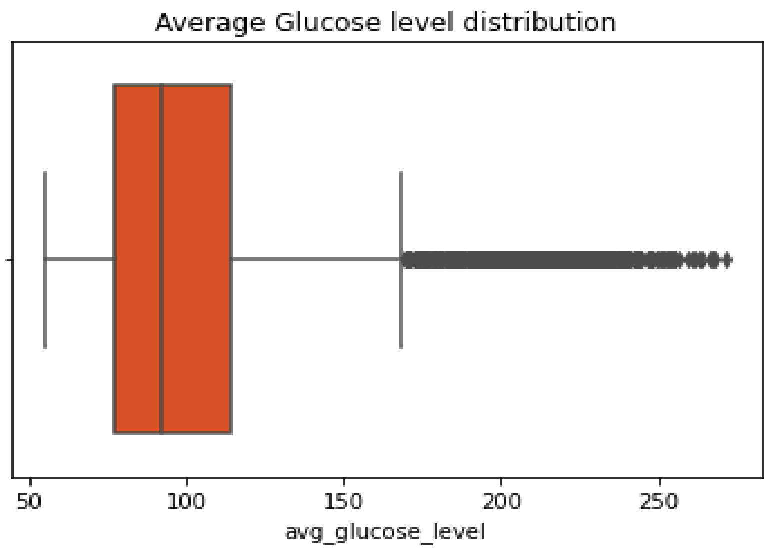
3.1.7. Detecting Outliers



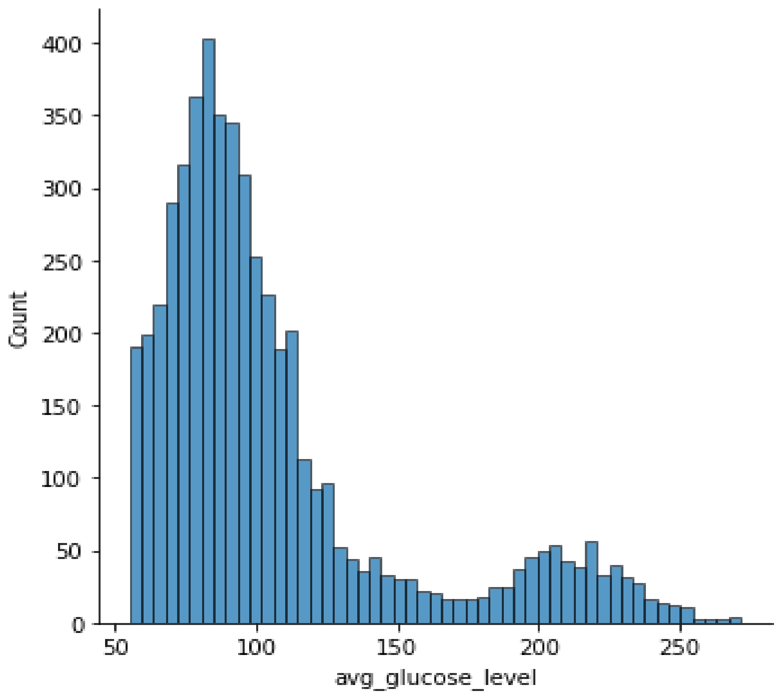
This plot shows the distribution of age of patients



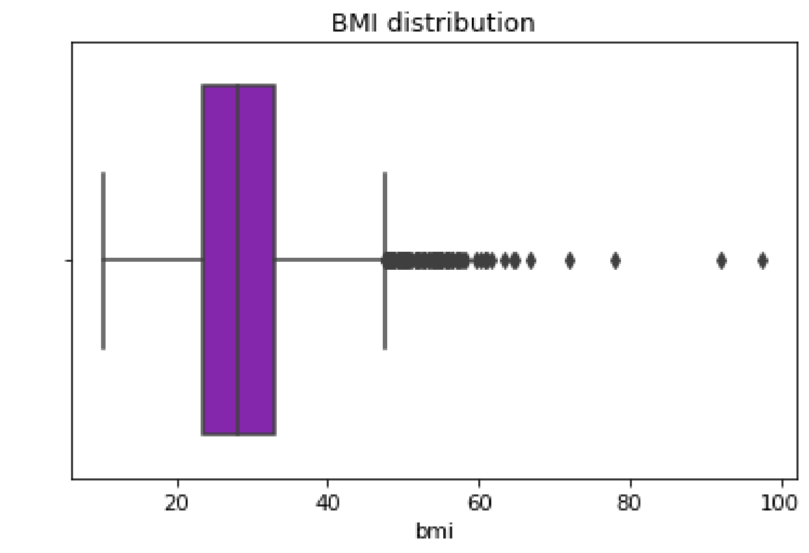
This plot shows the frequency of patients in the age groups



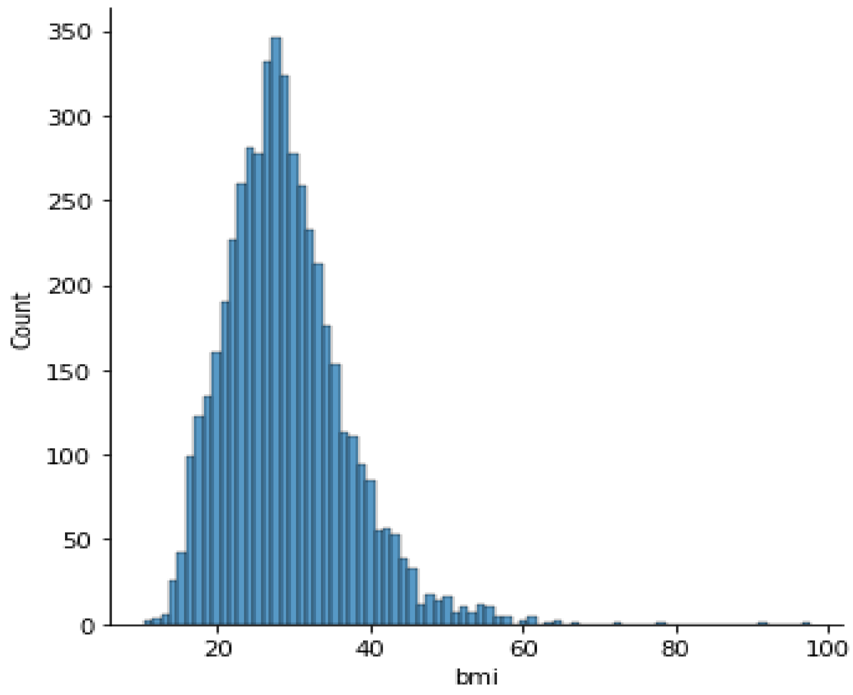
This plot shows the distribution of average glucose level in the patients



This plot shows the frequency of patients having different average glucose level



This plot shows the distribution of BMI of the patients



This plot shows the frequency of patients having different BMI

Chapter 4

DATA PREPROCESSING

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.[2]

When creating a machine learning project, we can not always come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So for this, we use data preprocessing tasks.[2]

Some general steps involved in data preprocessing are:[2]

1) Finding missing data and taking care of them.
2) Making sure that the data are in the right data types as needed for the work ahead.
3) Encoding categorical data.
4) Splitting the dataset into training and testing data.
5) Feature scaling.
6) Feature selection and conversion.
7) Data sampling and subsetting.

4.1 Handling Null Values

Null values are placed in a record when that attribute is missing or unknown. They can sometimes affect our computation because they represent a lack of data and our aggregate functions get a smaller amount of data to work with.

Null values can be there in the dataset for various reasons, like the person whose data is being collected is unwilling to share some information, or the data is just not available or discoverable for some reasons. Null values can also come if the data is corrupted.

There are quite a few ways to deal with Null values:[4]

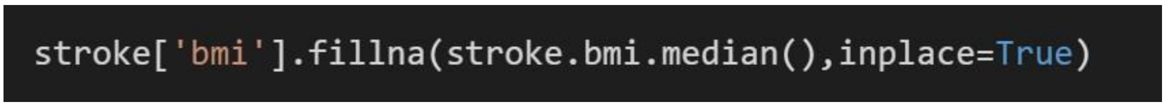
1) Deleting the records having null values. This is not always a good option because other attributes of the record may have important data, and in a dataset, many records can have null values, so removing the records would take away a big chunk of the original dataset and we will be left with a thin data collection.

2) Imputing missing values with mean, median or mode. Attributes that are numeric in general, that have missing values, can be imputed by taking the mean or median of the other values in the attribute. This method, though, does not factor in the covariance between variables.

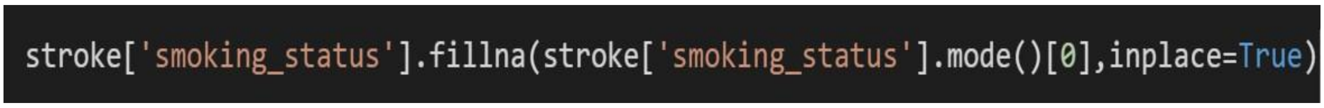
Attributes that are categorical can be imputed by taking the mode of the remaining values. Although this adds some new features to the model while encoding, which might affect the performance in the end.

3) Using methods like forward fill or backward fill. We can fill the missing values by the values which precede or succeed the missing values. Although this method might have its drawbacks as we might miss out on some important data factors.

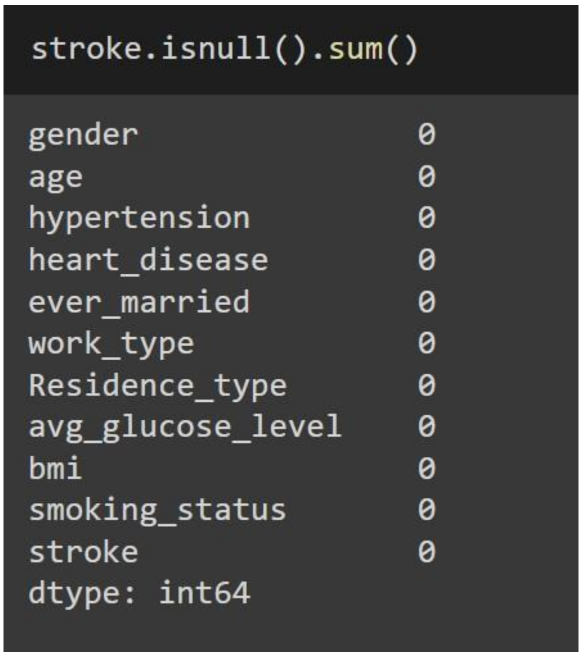
4) Predicting the missing values. We can try to predict the values which are missing using the existing data points, But if our dataset has a number of missing values, then there is not much data that the prediction algorithm can work on. Also, some datasets might be such that predicting the missing data is not an option.



```
stroke['bmi'].fillna(stroke.bmi.median(),inplace=True)
```



```
stroke['smoking_status'].fillna(stroke['smoking_status'].mode()[0],inplace=True)
```



```
stroke.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
dtype: int64
```

4.2 Feature Scaling

Feature Scaling is used to standardise the independent features in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. Without feature scaling, a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.[3]

4.2.1 Normalisation

Normalisation is a scaling technique in Machine Learning. It is applied during data preparation to change the values of numeric columns in the dataset to use a common scale. It is required only when features of machine learning models have different ranges, it is not necessary for all datasets.[2]

$$x_{\text{norm}} = \frac{x - min(x)}{max(x) - min(x)}$$

4.2.2 Standardisation

Standardisation scaling is also known as Z-score normalisation, in which values are centred around the mean with a unit standard deviation. This means the attribute becomes zero and the resultant distribution has a unit standard deviation. We can calculate the standardisation by subtracting the feature value from the mean and dividing it by standard deviation.[2]

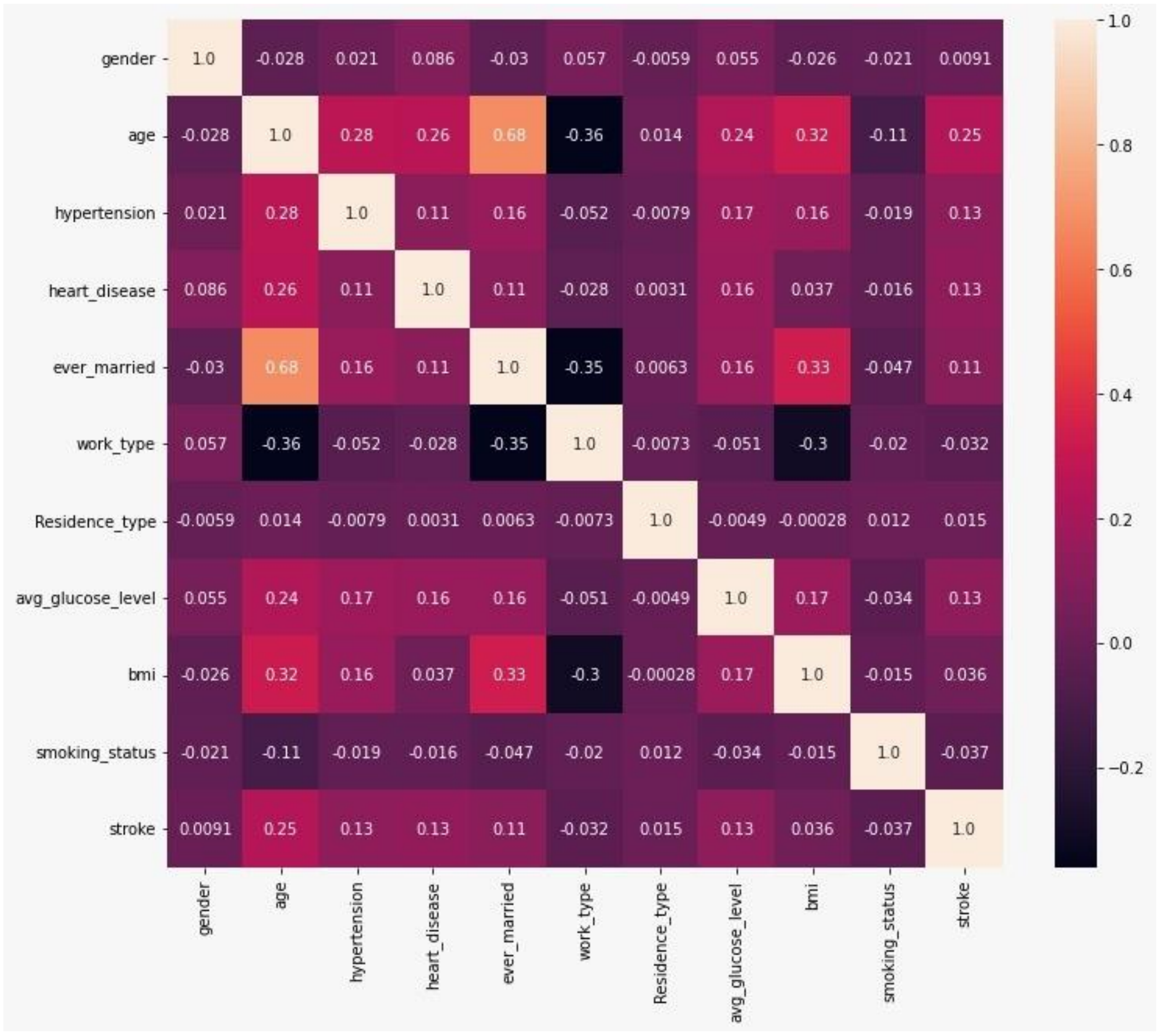
$$z = \frac{x - \mu}{\sigma}$$

Symbol	**Description**
Z	z-score
x	Data point in question
μ	Mean
σ	Standard deviation

4.3 Feature Selection and Conversion

It is a way to reduce dimensionality by using only a set of attributes. Any data is not lost in this way because only the redundant attributes are selected and dropped and the relevant attributes which help in the predictive modelling are kept.[1]

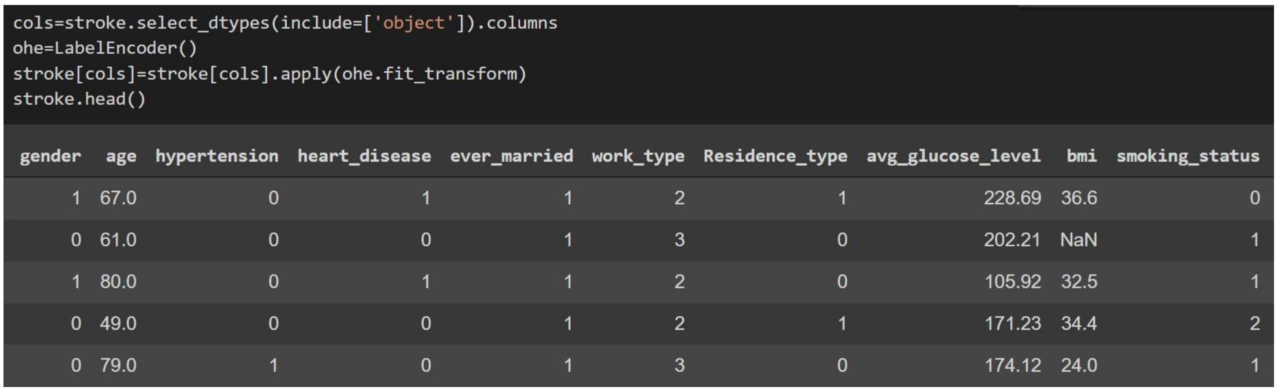
The attribute ‘id’ is not at all important in the predictive model, because it is a unique attribute. The attribute ‘age’ has the most influence on the target attribute ‘stroke’, then comes ‘heart_disease’, ‘avg_glucose_level’, ‘hypertension’ and ‘bmi’ in the decreasing order of influence.



	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
gender	1.0	-0.028	0.021	0.086	-0.03	0.057	-0.0059	0.055	-0.026	-0.021	0.0091
age	-0.028	1.0	0.28	0.26	0.68	-0.36	0.014	0.24	0.32	-0.11	0.25
hypertension	0.021	0.28	1.0	0.11	0.16	-0.052	-0.0079	0.17	0.16	-0.019	0.13
heart_disease	0.086	0.26	0.11	1.0	0.11	-0.028	0.0031	0.16	0.037	-0.016	0.13
ever_married	-0.03	0.68	0.16	0.11	1.0	-0.35	0.0063	0.16	0.33	-0.047	0.11
work_type	0.057	-0.36	-0.052	-0.028	-0.35	1.0	-0.0073	-0.051	-0.3	-0.02	-0.032
Residence_type	-0.0059	0.014	-0.0079	0.0031	0.0063	-0.0073	1.0	-0.0049	-0.00028	0.012	0.015
avg_glucose_level	0.055	0.24	0.17	0.16	0.16	-0.051	-0.0049	1.0	0.17	-0.034	0.13
bmi	-0.026	0.32	0.16	0.037	0.33	-0.3	-0.00028	0.17	1.0	-0.015	0.036
smoking_status	-0.021	-0.11	-0.019	-0.016	-0.047	-0.02	0.012	-0.034	-0.015	1.0	-0.037
stroke	0.0091	0.25	0.13	0.13	0.11	-0.032	0.015	0.13	0.036	-0.037	1.0

4.3.1 Categorical to Numerical

The attributes ‘gender’, ‘ever_married’,’work_type’, ‘Residence_type’, and ‘smoking_status’ are categorical attributes, so they have to be converted into their equivalent numerical values.



```
cols=stroke.select_dtypes(include=['object']).columns
ohe=LabelEncoder()
stroke[cols]=stroke[cols].apply(ohe.fit_transform)
stroke.head()
```

gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
1	67.0	0	1	1	2	1	228.69	36.6	0
0	61.0	0	0	1	3	0	202.21	NaN	1
1	80.0	0	1	1	2	0	105.92	32.5	1
0	49.0	0	0	1	2	1	171.23	34.4	2
0	79.0	1	0	1	3	0	174.12	24.0	1

4.4 Data Sampling and Subsetting

Splitting the dataset into training and testing data is necessary because the model to be created will learn from the training data and then use that knowledge to predict the outcome of the test data. Usually, the training data is much larger than the test data.

4.4.1 Sampling

Sampling is a commonly used approach for selecting a subset of the data objects to be analysed.[1]

Types of sampling are:[8]

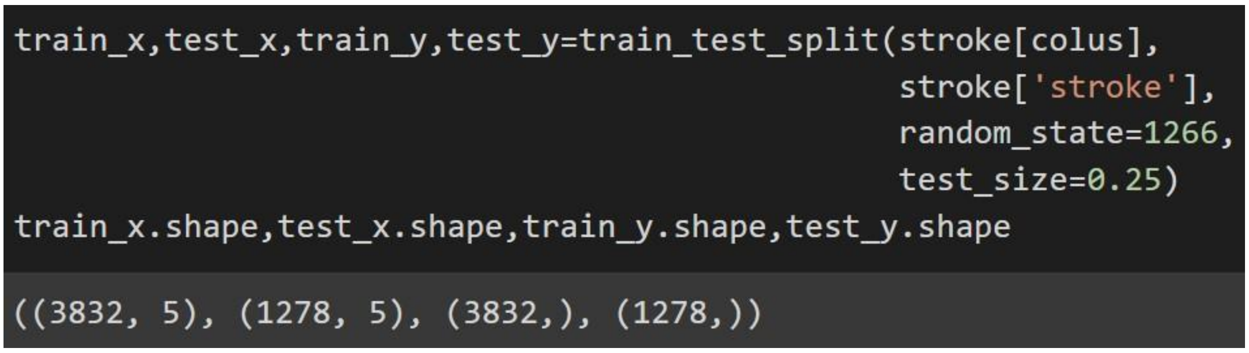
1. Simple random sampling: There is an equal probability of selecting any particular item.

2. Stratified sampling: Split the data into several partitions; draw random samples from each partition.

3. Sampling without replacement: As each item is selected, it is removed from the population.

4. Sampling with replacement: Objects are not removed from the population as they are selected for the sample (the same object can be picked up more than once).

4.4.2 TRAIN-TEST SPLIT



```
train_x,test_x,train_y,test_y=train_test_split(stroke[colus],
                                               stroke['stroke'],
                                               random_state=1266,
                                               test_size=0.25)
train_x.shape,test_x.shape,train_y.shape,test_y.shape
```

```
((3832, 5), (1278, 5), (3832,), (1278,))
```

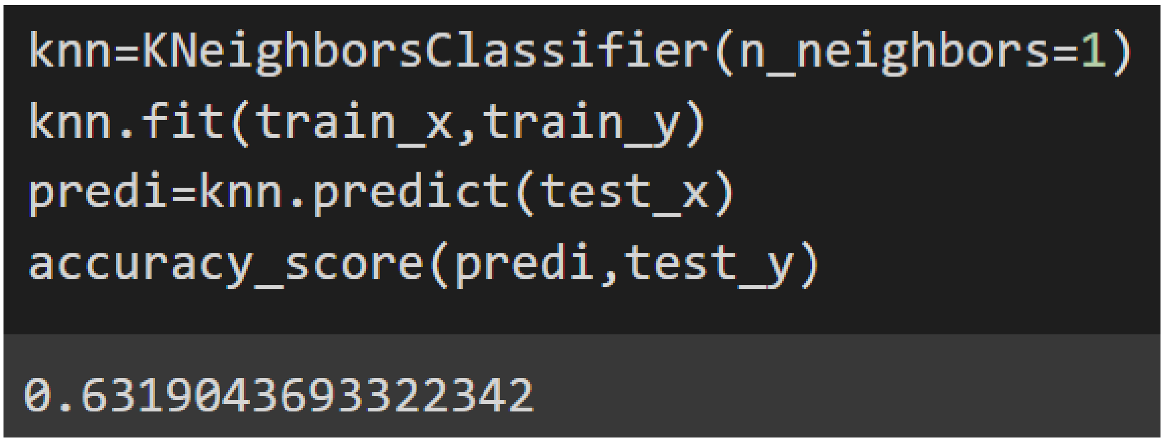
Chapter 5

BUILDING MODELS

Training and testing different models on the dataset is very important because different models will handle the data in different ways. The models should also be tinkered with by putting different parameters and thus finding out which gives the optimal result.

5.1 K-NN Model

K-Nearest Neighbour is a simple Machine Learning algorithm based on Supervised Learning technique. The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.[2]

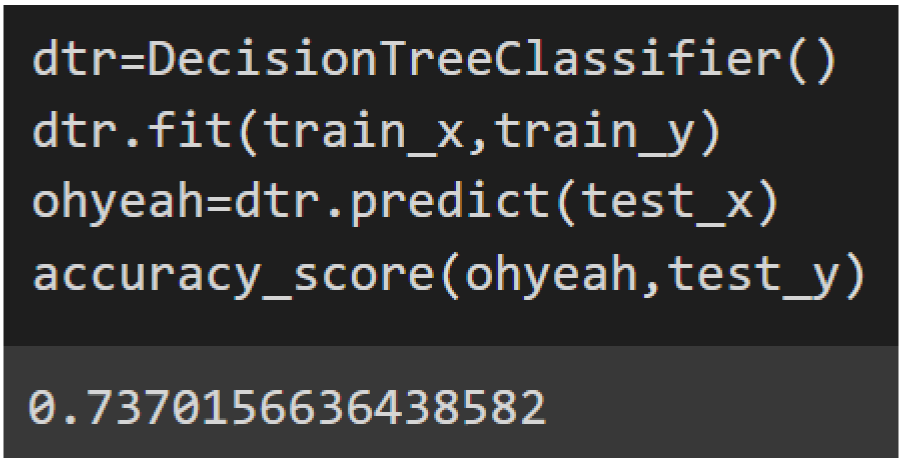


```
knn=KNeighborsClassifier(n_neighbors=1)
knn.fit(train_x,train_y)
predi=knn.predict(test_x)
accuracy_score(predi,test_y)
```

```
0.6319043693322342
```

5.2 Decision Tree Model

Decision Tree is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. It is based on the Supervised Learning technique.[2]

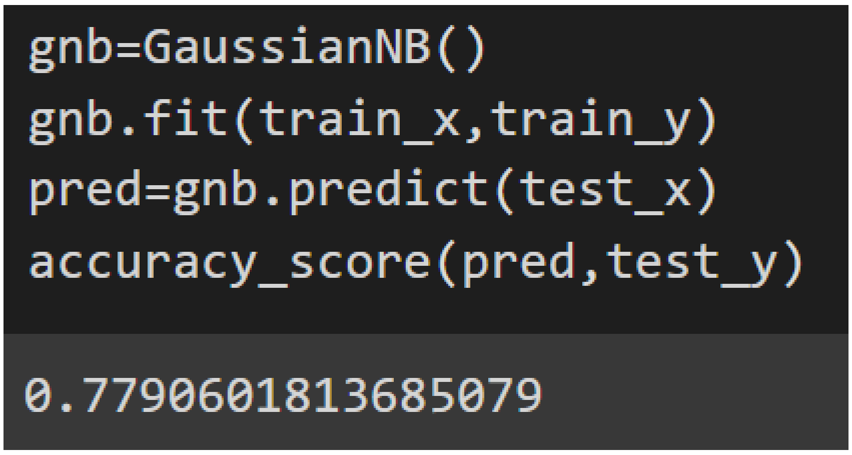


```
dtr=DecisionTreeClassifier()
dtr.fit(train_x,train_y)
ohyeah=dtr.predict(test_x)
accuracy_score(ohyeah,test_y)
```

```
0.737015636438582
```

5.3 Naïve-Bayes Model

Naive-Bayes algorithm is based on the Bayes theorem. It is a probabilistic classifier, it predicts on the basis of the probability of the object. It is based on the Supervised Learning technique.[2]

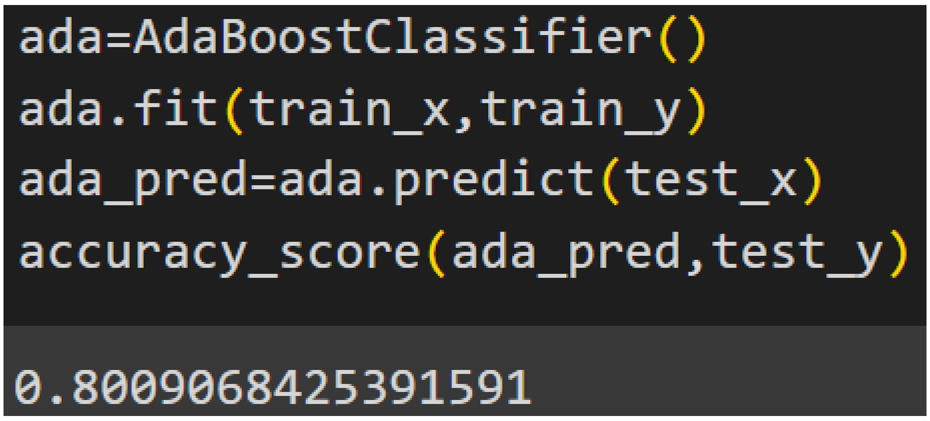


```
gnb=GaussianNB()
gnb.fit(train_x,train_y)
pred=gnb.predict(test_x)
accuracy_score(pred,test_y)
```

```
0.7790601813685079
```

5.4 Adaptive Boosting Model (AdaBoost)

AdaBoost, short for Adaptive Boosting, is a statistical classification meta-algorithm formulated by Yoav Freund and Robert Schapire in 1995, who won the 2003 Gödel Prize for their work.

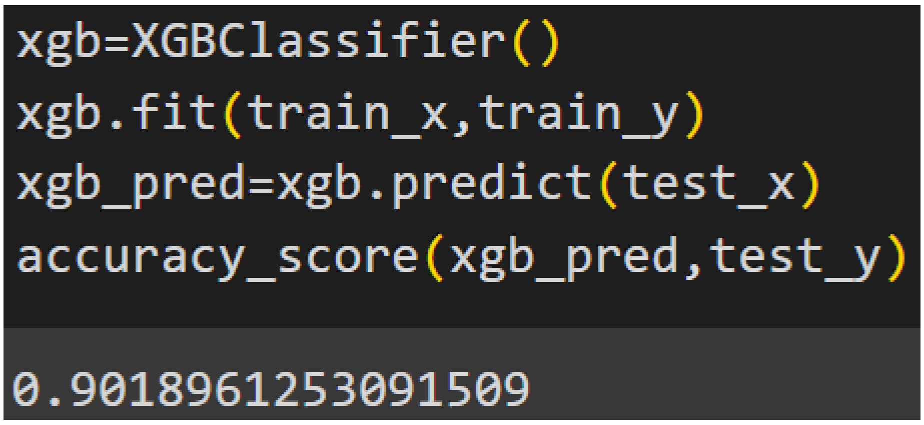


```
ada=AdaBoostClassifier()
ada.fit(train_x,train_y)
ada_pred=ada.predict(test_x)
accuracy_score(ada_pred,test_y)

0.8009068425391591
```

5.5 Extreme Gradient Boosting Model

XGBoost is an optimised distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework.



```
xgb=XGBClassifier()
xgb.fit(train_x,train_y)
xgb_pred=xgb.predict(test_x)
accuracy_score(xgb_pred,test_y)

0.9018961253091509
```

Chapter 6

MODEL EVALUATION AND RESULTS

6.1 Metrics

Data mining metrics are defined as a set of measurements that helps in determining the efficacy of a Data mining Method / Technique or Algorithm. They are important to help make the right decision, like choosing the right data mining technique or algorithm.[7]

6.1.1 Confusion Matrix

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known.[2]

n = Total Predictions	Actual: No	Actual; Yes
Predicted: No	True Negative	False Positive
Predicted: Yes	False Negative	True Positive

[2]

6.1.2 Accuracy

It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers.[2]

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

6.1.3 Precision

It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true.[2]

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

6.1.4 Recall

It is defined as the out of total positive classes, how our model predicted correctly. The recall must be as high as possible.[2]

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

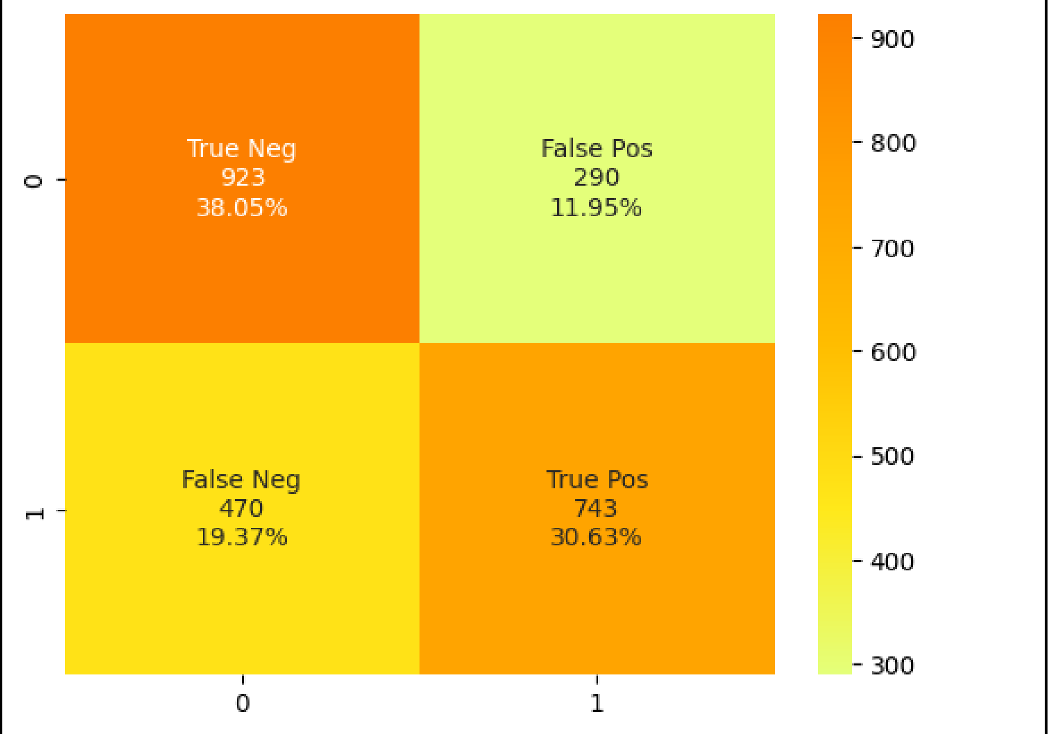
6.1.5 F1-Score

If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F1-score. F1-Score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision.[2]

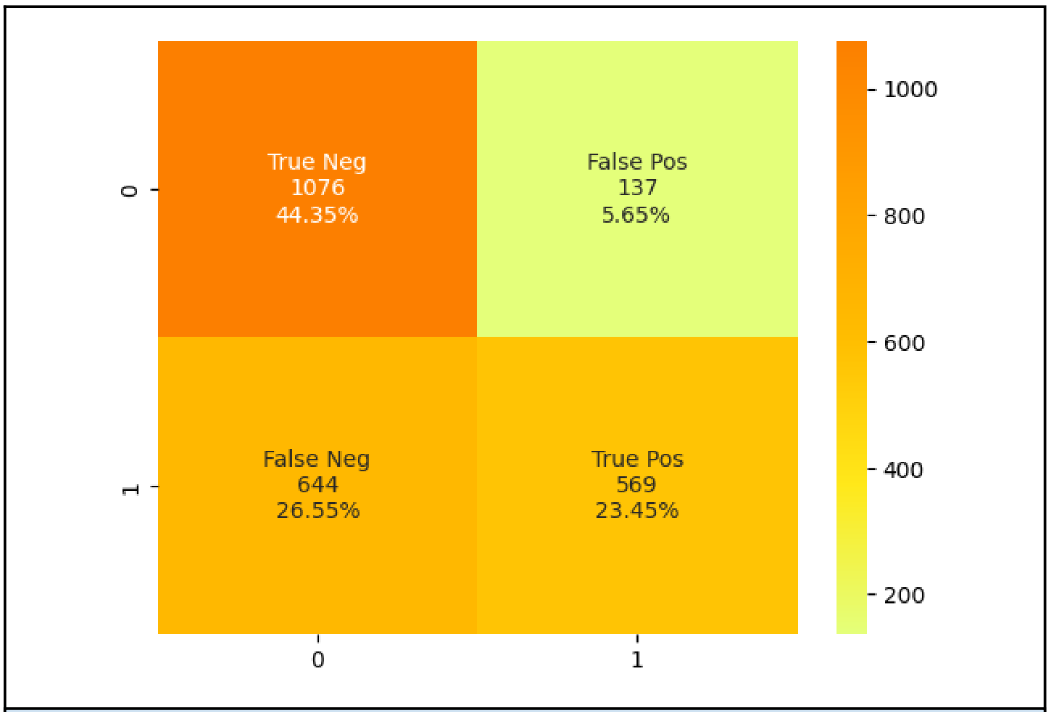
$$F1 - Score = \frac{2*Recall*Precision}{Recall + Precision}$$

6.2. Experimental Results and Comparison

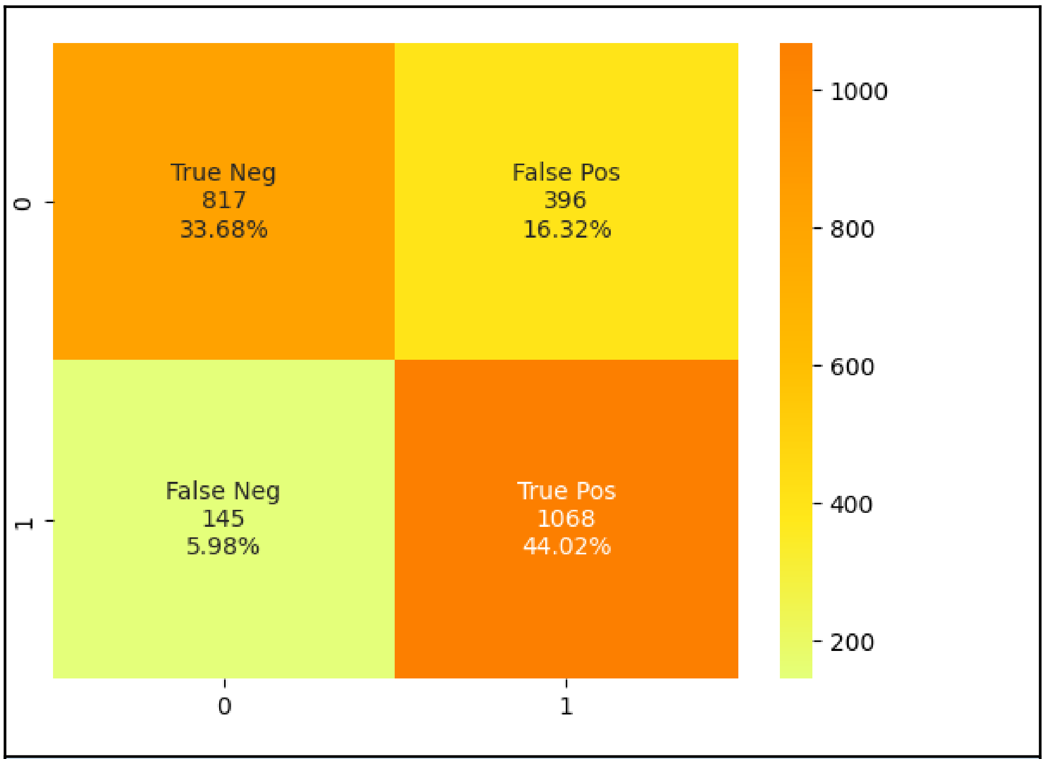
■ **Confusion Matrix**



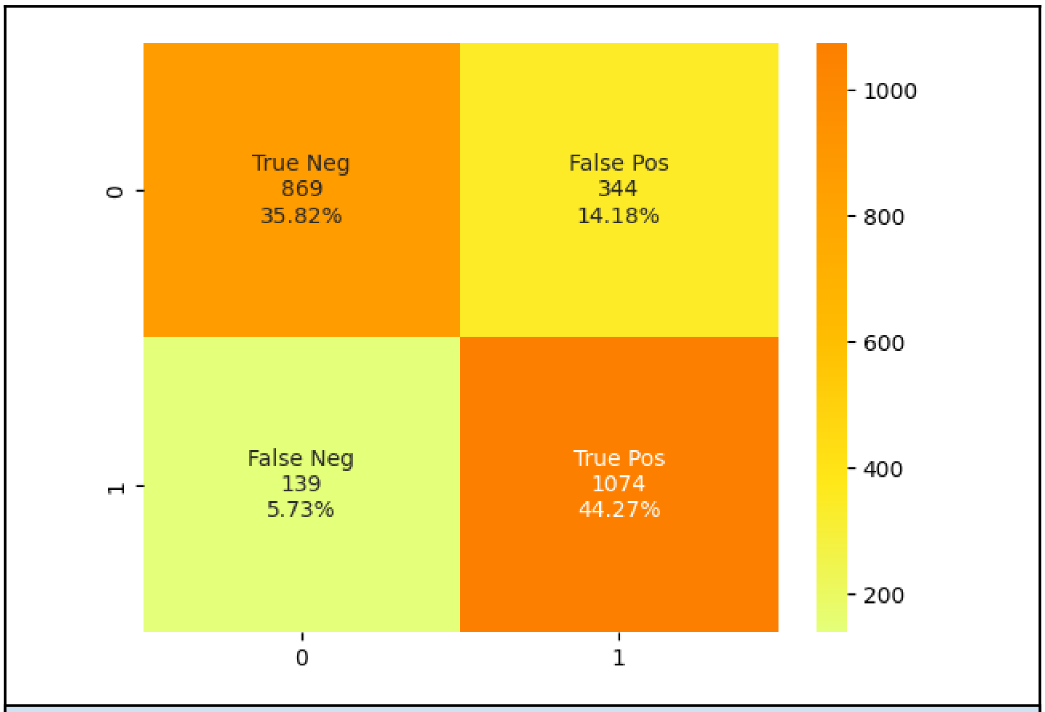
K-NN Model



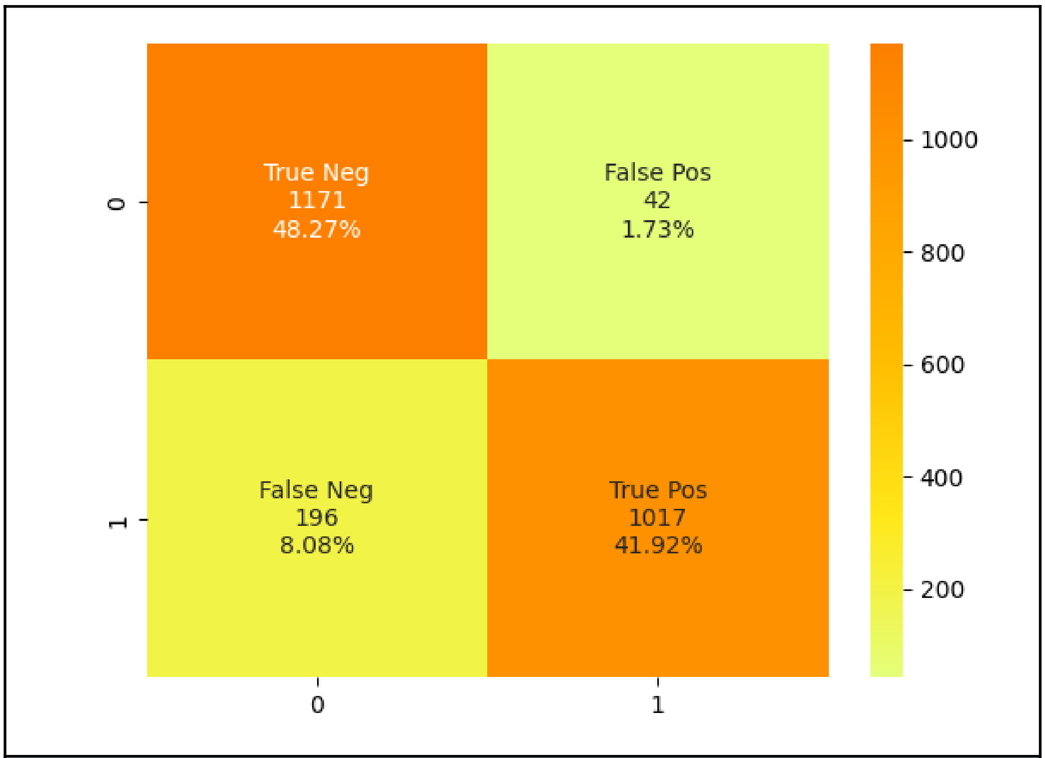
Decision Tree



Naive-Bayes



AdaBoost



XGBoost

- **Accuracy**

Models	Accuracy (%)
K-NN	63.1904369332
Decision Tree	73.7015663643
Naive-Bayes	77.9060181368
AdaBoost	80.0906842539
XGBoost	90.1896125309

- **Precision**

Models	Precision (%)
K-NN	72.4089635854
Decision Tree	85.4500616522
Naive-Bayes	73.1373889268
AdaBoost	75.7404795486
XGBoost	96.0339943342

- **Recall**

Models	Recall (%)
K-NN	42.6215993404
Decision Tree	57.1310799670
Naive-Bayes	88.2110469909
AdaBoost	88.5408079142
XGBoost	83.8417147568

- **F1-Score**

Models	F1-score (%)
K-NN	53.6585365853
Decision Tree	68.4782608695
Naive-Bayes	79.9701046337
AdaBoost	89.5246478873
XGBoost	81.6419612314

Chapter 7

INFERENCES AND CONCLUSION

Based on the findings in Chapter 6 the Naive-Bayes model has the highest accuracy among the other models used. Also in the confusion matrix the True negatives and the True positives are highest for the Naive Bayes model.

The attributes ‘age’, ‘hypertension’, ‘heart_disease’, ‘ever_married’, ‘avg_glucose_level’ have the highest influence on determining whether the patient will suffer from stroke or not, as written in the order.

References

1. Tan, P. N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.
2. Website: https://www.javatpoint.com
3. Website: https://www.geeksforgeeks.org/
4. Website: https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e#:~:text=Missing%20values%20can%20be%20handled,null%20can%20also%20be%20dropped.
5. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
6. Kesavaraj, G., & Sukumaran, S. (2013, July). A study on classification techniques in data mining. In *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)* (pp. 1-7). IEEE.
7. Website: himadri.cmsdu.org/documents/DataMining_Metrics.pdf

8. Website : https://online.datasciencedojo.com/course/Data-Mining/data-preprocessing-transformation/data-sampling-types

9. Website : kaggle.com
   Owner of dataset: fedesoriano