STROKE PREDICTION USING MACHINE LEARNING

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

##### 

##### 

##### 

##### 

##### **STROKE PREDICTION USING MACHINE LEARNING**

##### **A Major Project Report**

***in partial fulfilment for the award of the degree***

***of***

##### **BACHELOR’s OF TECHNOLOGY**

***in***

**COMPUTER SCIENCE ENGINEERING**

###### **by**

##### **MAYANK SINGH, 00420907218**

##### **PRITIKA RANA, 02420902717**

##### **SHIVAM RATHI, 03220902717**

**Guided by**

**Dr. SUNITA TIWARI**

**Assistant Professor, CSE**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**G.B. PANT GOVT. ENGINEERING COLLEGE, NEW DELHI.**

(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)

**MAY, 2021**

**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the Major Project entitled **“Stroke Prediction Using Machine Learning”** in partial fulfilment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering affiliated to **Guru Gobind Singh Indraprastha University, New Delhi** and submitted to the Department of Computer Science and Engineering G. B. Pant Govt. Engineering College is an authentic record of my work carried out during a period from March 2021 to June 2021. The matter represented in this report has not been submitted by me for the award of any other degree of this or any other institute/university.

**Date: - 14-07-2021 Name and Roll No:**

**Mayank Singh (00420907218)**

**Pritika Rana (02420902717)**

**Shivam Rathi (03220902717)**

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

#### *Signature of Supervisor*

**Date: - Dr. Sunita Tiwari**

**Assistant Professor, CSE**

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**

**LIST OF FIGURES i**

**ACKNOWLEDGMENT v**

**ABSTRACT vi**

**CHAPTER 1. INTRODUCTION 1**

1.1 OVERVIEW 1

1.2 MOTIVATION 6

1.3 PROBLEM STATEMENT 6

1.4 SCOPE OF PROJECT 6

1.5 ACCESS TO USERS 6

1.6 OUTLINE OF THE PROJECT 7

**CHAPTER 2. RELATED WORKS 8**

2.1 INTRODUCTION 8

2.2 EXISTING SYSTEM 8

2.3 PROPOSED SYSTEM 12

2.4 OUR APPROACH 12

**CHAPTER 3. PROBLEM DESCRIPTION & SPECIFICATION 13**

3.1 OVERVIEW 13

3.2 PROBLEM STATEMENT 13

3.3 SPECIFICATION 13

3.4 REQUIREMENTS 14

3.4.1 SOFTWARE REQUIREMENTS 14

3.4.2 HARDWARE REQUIREMENTS 15

3.4.3 OTHER REQUIREMENTS 15

**CHAPTER 4. SYSTEM DESIGN 16**

4.1 INTRODUCTION 16

4.1.1 DATA SOURCES 16

4.1.2 ML ALGORITHMS USED IN PROJECT 17

4.1.3 EVALUATION METRICS 17

4.2 ARCHITECTURE DIAGRAM 18

4.3 SYSTEM DIAGRAMS 19

4.3.1 USE CASE DIAGRAM 19

4.3.2 DATA FLOW DIAGRAM 20

4.4 SEQUENCE DIAGRAM 22

4.5 FLOW CHART DIAGRAM23

**CHAPTER 5. IMPLEMENTATION AND RESULT 24**

5.1 OVERVIEW 24

5.2 IMPORTING NECESSARY LIBRARIES 24

5.3 BASIC DATA ANALYSIS 25

5.4 PANDAS PROFILING 28

5.5 COORELATION 28

5.6 ANAMOLY DETECTION 31

5.7 HANDLING MISSING VALUES 33

5.8 ENCODING 35

5.8.1 LABEL ENCODING 36

5.8.2 ONE HOT ENCODING 36

5.9 SPLITING THE DATASET 37

5.9.1 STANDARDIZATION 37

5.9.2 SAVING THE SCALER OBJECT 38

5.10 COMPUTING TRAINING SCORE OF MODELS 39

5.11 MODEL SELECTION & OPTIMIZATION 47

5.11.1 SAVING THE BEST MODEL 48

5.11.2 K-FOLD CROSS VALIDATION 48

5.11.3 HYPER PARAMETER OPTIMIZATION 49

5.11.4 DATA FLOW DIAGRAM 50

5.12 WEB APP DEVELOPMENT 51

5.13 CONNECTING FRONTEND & BACKEND52

5.14 PREDICTING THE RISK OF STROKE 53

**CHAPTER 6. CONCLUSION AND FUTURE SCOPE 55**

6.1 CONCLUSION 55

6.2 FUTURE SCOPE55

**CHAPTER 7. REFERENCES 56**

**APPENDICES 62**

**APPENDIX 1: USER GUIDE 62**

**List of Figures**

**Figure No. Figure Title Page No.**

|  |  |  |
| --- | --- | --- |
| Figure 1.1 | Figure showing blockage in tissue, which happens in the case of ischemic stroke | 2 |
| Figure 1.2 | Figure representing the difference in TIA and Ischemic Stroke | 2 |
| Figure 1.3 | Figure representing rapture of tissue as happens in hemorrhagic stroke | 3 |
| Figure 1.4 | Stroke deaths per million persons in 2012 | 4 |
| Figure 1.5 | [Disability-adjusted life year](https://en.wikipedia.org/wiki/Disability-adjusted_life_year) for cerebral vascular disease per 100,000 inhabitants in 2004. | 5 |
| Figure 3.1 | Process Flow Diagram | 14 |
| Figure 4.1 | Output showing the dataset information | 17 |
| Figure 4.2 | Architecture of the Stroke prediction model | 19 |
| Figure 4.3 | Use Case Diagram for stroke prediction system | 19 |
| Figure 4.4 | DFD level 0 | 20 |
| Figure 4.5 | DFD level 1 | 21 |
| Figure 4.6 | DFD level 2 | 21 |
| Figure 4.7 | Sequence Diagram of the system | 22 |
| Figure 4.8 | Flow chart for the stroke prediction system | 23 |
| Figure 5.1 | Code snippet showing the import of libraries used in the project | 24 |
| Figure 5.2 | Code snippet to read the dataset | 24 |
| Figure 5.3 | 10 Sample data from the dataset | 25 |
| Figure 5.4 | Code snippet and resultant table after dropping the id column | 25 |
| Figure 5.5 | Code snippet to visualize the dataset | 26 |
| Figure 5.5a | Pie chart representing the distribution of smoking status of individuals in percentage in the dataset | 26 |
| Figure 5.5b | Boxenplot on the dataset showing relation of stroke with age, BMI and average glucose level | 27 |
| Figure 5.5c | Figure representing swarm of individual records representing relation between age, gender, marital status, smoking habits, work type and likelihood of stroke | 27 |
| Figure 5.6 | Code snippet showing pandas profiling | 28 |
| Figure 5.6a | Pandas profile report | 28 |
| Figure 5.7 | Code snippet to show correlation | 29 |
| Figure 5.8 | Sample output showing the heatmap of correlation | 29 |
| Figure 5.9 | Code snippet for pairplot representing how different attributes is related to each other | 30 |
| Figure 5.10 | Sample output showing the distribution of values for each attribute in their domains, their relation with other attribute and how they play role in stroke risk | 31 |
| Figure 5.11 | Code snippet showing the detection of outliers | 32 |
| Figure 5.12 | Output showing the detection of outliers | 32 |
| Figure 5.13 | Code snippet to drop the dataset containing outliers | 32 |
| Figure 5.14 | Resultant table after anomaly detection and rectification | 33 |
| Figure 5.15 | Code snippet to check for missing bmi values | 33 |
| Figure 5.16 | Output showing the rows where BMI value is null | 33 |
| Figure 5.17 | Output showing the boxplot grouped by gender | 34 |
| Figure 5.18 | Code snippet and output showing the mean BMI based on gender | 34 |
| Figure 5.19 | Code snippet and output showing that none of the field contains null or missing values | 34 |
| Figure 5.20 | Code snippet for encoding | 35 |
| Figure 5.21 | Code snippet and output for handling categorial values | 35 |
| Figure 5.22 | Code snippet to implement label encoding | 36 |
| Figure 5.23 | Code snippet to implement one hot encoding | 37 |
| Figure 5.24 | Sample output showing the table doesn’t contain categorial variables | 37 |
| Figure 5.25 | Code snippet showing the dataset is split for the purpose of training, testing and validation | 38 |
| Figure 5.26 | Output showing the amount of data used for training, testing and validation | 38 |
| Figure 5.27 | Code snippet to implement Standardization | 38 |
| Figure 5.28 | Code snippet for saving the scaler object | 39 |
| Figure 5.29 | Code snippet for computing accuracy, precision, recall, F1 score, specificity and confusion matrix of various ML algorithms | 40 |
| Figure 5.30 | Output containing accuracy, precision, recall, F1 score, specificity and confusion matrix of various ML algorithms | 45 |
| Figure 5.31 | Code snippet to graphically visualize the train-validation-test score of models | 45 |
| Figure 5.32 | Graphical output of train-validation- test score of models | 46 |
| Figure 5.33 | Code snippet and output showing the accuracy of each model | 46 |
| Figure 5.34 | Comparison Score of different models representing scores of accuracies, precision, recall, and F1 score | 47 |
| Figure 5.35 | Code snippet for model selection and training for optimization | 47 |
| Figure 5.36 | K Neighbors Classifier score along with accuracy and confusion matrix | 48 |
| Figure 5.37 | Code snippet for saving the KNN Model | 48 |
| Figure 5.38 | Code snippet to implement K-Fold Cross validation | 48 |
| Figure 5.39 | Graphical output after implementing K-Fold Cross validation | 49 |
| Figure 5.40 | Code snippet and sample output after applying GridSearchCV | 50 |
| Figure 5.41 | Code snippet and sample output after applying RandomizedSearchCV | 50 |
| Figure 5.42 | Code snippet for best feature selections | 51 |
| Figure 5.43 | Sample output showing train, validation and test dataset | 51 |
| Figure 5.44 | Layout of the webpage (User form) | 52 |
| Figure 5.44a | Sample input data filled in the Form | 53 |
| Figure 5.44b | Sample results displayed at the Web Platform when using the model for prediction | 54 |

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our supervisor’s **Dr Sunita Tiwari** & **Dr Anu Saini** for providing their invaluable guidance, comments and suggestions throughout the project. We would specially thank Dr Sunita Tiwari for constantly motivating us to work harder and suggesting us different ways to tackle the problems encountered.

Also, we would like to appreciate the whole team effort for the completion of this project. We would like to express our gratitude for the Department of Computer Science Engineering, Gobind Ballabh Pant Government Engineering college for providing us with the opportunity to work on this project and to all the teachers and mentors guiding along the way.

* Mayank Singh (00420907218)
* Pritika Rana (02420902717)
* Shivam Rathi (03220902717)

**ABSTRACT**

The project Stroke Prediction Using Machine Learning provides user an ability to check the chances of stroke to him/her, based on different parameters asked to him/her. The user interacts with the project with the help of a Web Platform which contains a form having different key parameters which constitutes in stroke. On providing these inputs the model predicts the chances of stroke to him/her.

Stroke is a deadly disease and a leading cause of death. In most of the case the patient suffers grave consequences, along with the patient the family and relatives also suffer a lot. But studies suggest that 80 percent of stroke can be pre-diagnosed and, in that scenario, patient can be saved. There are some key parameters which contributes to chances of stroke. There parameter includes age, gender, hypertension, blood pressure, heart disease, average glucose level, BMI, smoking habits, work type, place of residence, marital status, previous history of stroke or heart attack etc. of the patient. As a patient is mostly well aware of these factors, he/she can get a diagnosis in the case of suspicion about chances of stroke. A lot of machine learning methods are developed and are currently under research to predict the chances of stroke based on different parameters.

In this project we have taken 10 different machine learning classifiers and tested them on our dataset “Stroke dataset” for accuracy. Then selected the best one for prediction. We have also performed k-fold validation and hyper-parameter optimization on the model. Along with that we have tried to identify the best features for the classifiers. The inputs gathered from user via the web platform is provided to the model using the flask backend system and the saved model is used for predicting the output which is displayed to the user on the result page.

The project provides user the facility to check the chances of stroke to them based on inputs features at the choice of location and time through their web enabled devices. The project will be hosted online on web app or mobile app and can be used by user from anywhere. Along with this our objective is also to compare various machine learning algorithm and learn how they perform on the dataset for predicting the stroke risk. What are differences among different model and also understand which classifiers performs best and what may be the reasons behind it.

**CHAPTER-1**

**INTRODUCTION**

* 1. **OVERVIEW**

# **1.1.1 Stroke:**

A stroke happens when the blood supply to a piece of your brain is hindered or decreased, forestalling mind tissue from getting oxygen and supplements. Brain cells being to die in minutes. Stroke is a health-related crisis, and prompt treatment is significant. Early activity can diminish brain harm and different intricacies. In 2018, 1 in each 6 deaths from cardiovascular illness was because of stroke in USA. Consistently, in excess of 795,000 individuals in the United States have a stroke.[1]

Signs and indications of a stroke may incorporate a failure to move or feel on one side of the body, issues understanding or talking, dizziness, or loss of vision to one side. Signs and side effects frequently appear soon after the stroke has occurred. In the event that manifestations last short of what a couple of hours, the stroke is a transient ischemic assault (TIA), likewise called a mini stroke. A hemorrhagic stroke may likewise be related with an extreme headache. The indications of a stroke can be lasting. Long haul complexities may incorporate pneumonia and loss of bladder control.[2]

There are two main types of stroke:

**Ischemic stroke**

This is the most widely recognized sort of stroke, making up 87% of all cases. A blood coagulation keeps blood and oxygen from arriving at an area in brain. It happens when the mind's veins become limited or hindered, causing seriously decreased blood stream (ischemia) as displayed in Figure 1.1. Obstructed or limited veins are brought about by fatty deposits that develops in veins or by blood clumps or other debris that travel through your circulatory system and lodge in the veins in your brain.[2]

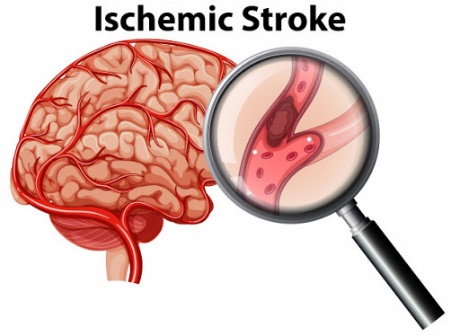


Figure 1.1 Figure showing blockage in tissue, which happens in the case of ischemic stroke

Factors contributing in ischemic stroke are:

* Obstruction of a vein by a blood coagulation formed locally
* Block because of an embolus from somewhere else in the body,
* General decrease in blood supply, e.g., in [shock](https://en.wikipedia.org/wiki/Shock_(circulatory))

An important type of ischemic stroke is **transient ischemic stroke**. This happens when blood stream to a piece of the mind is lacking for a short span of time. It is a transitory time of indications like those you'd have in a stroke. Typical blood flow resumes after a short time, and the side effects resolve without treatment. They are also called a ministroke. A TIA doesn't cause lasting harm. They're brought about by a transitory lessening in blood supply to part of your brain, which may last just five minutes. Figure 1.2 shows how ischemic stroke is not the same as transient ischemic attack.[2]

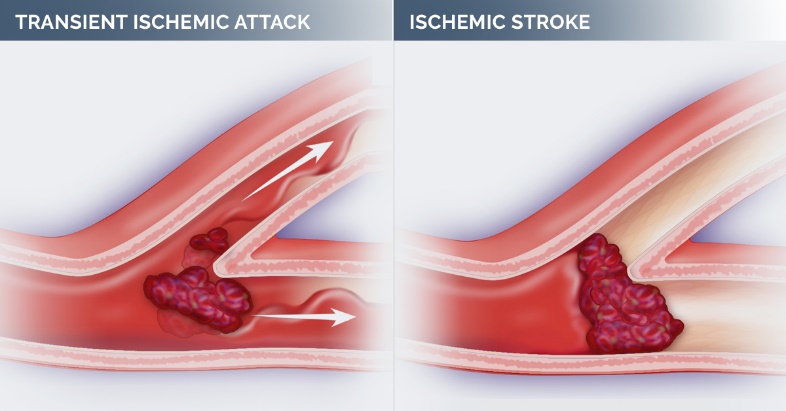


Figure 1.2 Figure representing the difference in TIA and Ischemic Stroke

**Hemorrhagic stroke**

Hemorrhagic stroke happens when a vein in your mind breaks or cracks as displayed in Figure 1.3. Brain hemorrhages can result from numerous conditions that influence your veins. These are generally the aftereffect of aneurysms (an aneurysm refers to a weakening of an artery wall that creates a bulge, or distention, of the artery). [2]

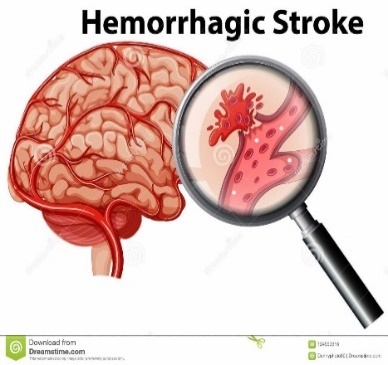


Figure 1.3 Figure representing rapture of tissue as happens in hemorrhagic stroke

Factors leading to hemorrhagic stroke include:

* Extremely high blood pressure
* Overtreatment with blood thinners (anticoagulants)
* Bulges at weak spots in your blood vessel walls (aneurysms)
* Trauma (such as a car accident)
* Protein stores in vein walls that lead to fragility in the vessel wall (cerebral amyloid angiopathy)
* Ischemic stroke leading to hemorrhage

An ischemic stroke, whenever distinguished inside three to four and half hours, might be treatable with a prescription that can separate the coagulation. Some hemorrhagic strokes profit with a medical procedure. Treatment to endeavor recuperation of lost capacity is called stroke recovery, and preferably happens in a stroke unit; nonetheless, these are not accessible in a significant part of the world.[2]

### In 2013, roughly 6.9 million individuals had an ischemic stroke and 3.4 million individuals had a hemorrhagic stroke. In 2015, there were about 42.4 million individuals who had recently suffered a heart attack were as yet alive. Somewhere in the range of 1990 and 2010 the quantity of strokes which happened every year diminished by around 10% in the created world and expanded by 10% in the creating scene. In 2015, stroke was the second most regular reason for death after coronary course sickness, representing 6.3 million deaths (11% of the aggregate). About 3.0 million passing came about because of ischemic stroke while 3.3 million deaths came about because of hemorrhagic stroke. About portion of individuals who have had a stroke live short of what one year. Generally speaking, 66% of strokes happened in those more than 65 years of age.[2]

### **General risk factors which build the other factors which causes different types of stroke are:**

* High blood cholesterol levels,
* High blood pressure and atrial fibrillation.
* Cigarette smoking (active and passive),
* Heavy [alcohol](https://en.wikipedia.org/wiki/Alcohol_consumption_and_health) use and/or drug use,
* Risk factor increases by a factor of 30% by smoking a cigarette a day,
* [Diabetes mellitus](https://en.wikipedia.org/wiki/Diabetes_mellitus),
* Lack of [physical activity](https://en.wikipedia.org/wiki/Physical_activity), [obesity](https://en.wikipedia.org/wiki/Obesity), processed [red meat](https://en.wikipedia.org/wiki/Red_meat) consumption, and unhealthy diet,
* [End-stage kidney disease](https://en.wikipedia.org/wiki/End-stage_kidney_disease).[2]

## **Epidemiology [2]**

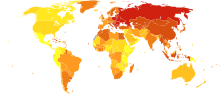
[](https://en.wikipedia.org/wiki/File:Stroke_world_map-Deaths_per_million_persons-WHO2012.svg)

Figure 1.4 Stroke deaths per million persons in 2012

  58–316   576–640

  317–417   641–771

  418–466   772–974

  467–518   975-1,683

  519–575   1,684–3,477

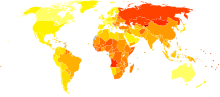
[](https://en.wikipedia.org/wiki/File:Cerebrovascular_disease_world_map_-_DALY_-_WHO2004.svg)

Figure 1.5 [Disability-adjusted life year](https://en.wikipedia.org/wiki/Disability-adjusted_life_year) for cerebral vascular disease per 100,000 inhabitants in 2004.

|  |  |
| --- | --- |
| no data    <250    250–425    425–600    600–775    775–950    950–1125 | 1125–1300    1300–1475    1475–1650    1650–1825    1825–2000    >2000 |

Stroke was the second most incessant reason for death worldwide in 2011, representing 6.2 million deaths (~11% of the aggregate). Around 17 million individuals had a stroke in 2010 and 33 million individuals have recently suffered a heart attack were as yet alive. By and large, 66% of strokes happened in those more than 65 years of age. South Asians are at especially high danger of stroke, representing 40% of worldwide stroke losses. The danger of stroke increments dramatically from 30 years old, 95% of strokes happen in individuals age 45 and more seasoned, and 66% of strokes happen in those beyond 65 years old. An individual's danger of a stroke likewise increments with age.[2]

Family members may have a genetic tendency for stroke or offer a lifestyle that adds to stroke. The aftereffects of this investigation tracked down that the lone huge hereditary factor was the individual's blood classification. Having had a stroke in the past significantly builds one's danger of future strokes. [2]

Men 25% more likely to suffer strokes than women, yet 60% of deaths from stroke occur in women. Some risk factors for stroke apply just to women. Fundamental among these are pregnancy, work, menopause, and the treatment thereof (HRT).[2]

**1.2 MOTIVATION**

The stroke prediction model provides the chances of stroke to the user. This is done by collecting the inputs from user from the web platform and use the trained model to predict the chance of stroke to the user. It is easy to use and always available and can be accessed from anywhere and anytime through the web enabled device. Whenever a person feels symptoms of stroke like loss in vision in one eye, inability to move or feel on one side of the body, problem in understanding or speaking, dizziness these could be symptoms of stroke. There are some key risk factors like blood pressure, smoking habits, high blood cholesterol, obesity, diabetes, kidney problem etc. which are often know to user. In this case he/she can do a self-diagnosis test for chances for stroke risk to him/her and then consult a doctor. Timely detection can help to a large extend and can even save life in some instances. The easy to access app will provide ease to user. Also, the project will help us understand the capability and factors useful for machine learning classifiers for predicting such results.

**1.3 PROBLEM STATEMENT**

To compare various machine learning classifiers based on accuracy and select the best one to build a system for predicting stroke risk.

**1.4 SCOPE OF PROJECT**

This project provides user with an easy to access app which they can use to check the chances of stroke to them. The aim of the project is to make such app available to everyone. This project also plans to learn which machine learning classifiers are best suited for this kind of job and understand the reason behind it.

**1.5 ACCESS TO USERS**

The project aims to host the web app or mobile app for the users making it easily available for everyone at all time. The interface aims to be very basic and user friendly. The major objective is this app can be used by users on phone instantaneously and does not require any expertise.

**1.6 OUTLINE OF THE PROJECT**

The following chapters in this report give a detailed glance of effort put in the direction of development of this project. Chapter 2 contains some of the work related to stroke prediction. It outlines some of the key works done in the field of stroke prediction. These works lay the ground to the work that we have persuaded in our project. Chapter 3 contains the description of our project and contains information about the specification of the project and its hardware and software requirements.

Chapter 4 contains the system design, details about the algorithms implemented in our project, the parameters on which our project is based upon and how the data flow works in the project. With the help of different diagram, we have tried to explain the approach we have employed in our project. We have also given a brief introduction to metrices we have used for testing and comparing various ML classifiers.

Chapter 5 contains the implementational details of our project work and snippets of program code and working of the project along with some of the results. We have tried to follow a structured approach throughout the development and displayed the same in this chapter. Chapter 6 represents the conclusion we have achieved through this project and also highlights some of the future scope that we believe will be a great enhancement to our project.

**CHAPTER – 2**

**RELATED WORK**

**2.1 INTRODUCTION**

In this section, we will look into some of the earlier works presented in the direction of this project. There has been a lot of work that has already happened in the field of prediction in the medical department. While preparing this project we have gathered ideas from different research materials and models. Here, we are going to enlist them.

**2.2 EXISTING SYSTEM**

Lately a lot of research has been done on predicting strokes. The research can be briefly categorized into three different categories including the dataset collection, algorithm comparison and efficient & accurate stroke rate prediction.

AI techniques for assessment of neuro imaging data are used to assist in diagnosis of stroke. Conclusion and treatment of stroke illness is complex, and developing countries because of the shortfall of analytic gadgets just as a deficiency of specialists and numerous different assets that influence satisfactory expectation and medication of stroke patients. As of late, computer innovation and AI strategies are confronting this worry, to work on the framework to help specialists in the primer stage in settling on choices about illness.[8] Numerous frameworks have been created as of late to for stroke forecasts varying by little factors. Some of which we covered are examined underneath.

Yu et al. [52] have implemented the machine learning techniques by considering the decision tree algorithm of C4.5. The proposed methodology of this work uses 13 features rather than 18 stroke scale features for determining and analysing the stroke classification. The data collected from the database of the National Institutes of Health Stroke Scale (NIHSS) for the study of cerebrovascular strokes among people affected over age 65 years. Out of total samples, 75% of subjects were used for training, and 25% of subjects were used for testing. Based on the hypothetical solutions 91.11% decent accuracy is obtained through the decision tree algorithm.

Monteiro et al. [53] have implemented a machine learning methodology to determine the practical results of the patients affected with ischemic stroke when admitted for three weeks. Among different types of strokes, Ischemic stroke acts as a major purpose of disorder and death all over the world among people of 65 years and in adults. The proposed methodology is succeeded on an outcome of the outlined superior AUC value of 0.808± 0.085 when compared to the foremost point score of 0.771 ± 0.056 with 70% subjects used for training and 30% subjects used for testing. On the other hand, the model keeps on increasing the additional features depending upon particular timing along with the increase in the AUC score by setting the point score of above 0.90. The Baseline feature sets used under experiment -1 produced a ‘good’ outcome with 51.3% and a ‘poor’ outcome with 48.7% accuracies on 425 samples. By obtaining the conclusions and validating the results taken at the time of admission and by making a priority of the use of technological methods whenever required.

Sung et al. [54] have proposed a methodology that can be examined for automated phenotyping by further classifying the ischemic stroke into 4 subdivisions. This model depending upon the structured and unstructured data taken from the electronic medical records (EMRs). It works on the records of 4640 patients who have been diagnosed with the mild symptoms of Ischemic stroke and also been taken for examining the results. The sub-divisions structured data has National Institutes of Health stroke scale whereas unstructured data has clinical narratives which are refined through a heatmap. The conclusion of stroke scale data from EMRs could make the process clear and smooth phenotyping of ischemic stroke when integrated with the structures data. However, diminishing the different levels of class issues into binary classification work along with the congregation of classifies solution helps in increasing the performance by taking 66% subjects on training and 34% subjects on testing.

Xie et al. [55] have proposed a model to combine common stroke biomarkers by developing machine learning techniques and to analyse the complete recovery of the ischemic stroke patient within three months. In this work, to predict the recovery terms of the patient Extreme gradient boosting (XGB) and Gradient Boosting Machine (GBM) models were implemented to identify modified ranking scale (mRS) scores by using biomarkers availability within 24 hours of the admitting of the patient. A total of 512 patients records were taken into consideration for analysis with fivefold cross-validation for identifying the improvements of the model. These records are categorized into 80% on training and 20% on testing. Under the binary analysis of an mRS score which is larger than 2 considering biomarkers which are provided during the time of admitting, XGB and GBM include AUC of scores 0.746 and 0.748 accordingly.

Wang et al. [56] have implemented a machine learning model in the configuration of the risk of symptomatic intracerebral haemorrhage (sICH) after the thrombolysis of the stroke. The risk factors of sICH are theoretically used after stroke thrombolysis. Based on this study, a total of 2578 thrombolysis-treated ischemic stroke patients were recognized from January 2013 and December 2016. Out of which 70% were taken into training modules and 30% considered under nominal data test sets. In order to analyse the risk of sICH, these machine learning modules were helped to increase the performance analysis metrics through the area under curve (AUC) with 0.82.

Lin et al. [57] have proposed a hybrid neural network model with 10 cross folds for evaluating the stroke outcome. The data collected from “Taiwan Stroke Registry” is given for the model with 70% on training and 30% on testing.

Sung et al. [58] have implemented machine learning algorithms to analyse the stroke outcome with acute minor stroke. Among 739 patients, 61 patients having a negative outcome after a stroke at 90 days. The data is categorized into 89.4% for no END and the remaining 10.6% for END This database related to patients was taken from NIHSS with a score of ≤ 3. Pre indication of the neurological deterioration tells us that diminishing of the NIHSS score within days of the admission of the patient. The inimical score was determined from the modified Ranking scale score of ≥ 2. In this work, four machine learning models such as bootstrap decision forest, boosted trees, Logistic Regression, and Deep neural network was used in analysing the early signs of neurological deterioration and examined with a decent accuracy of 94.6%.

Govindarajan et al. [60] led a study to classify stroke problem utilizing a text mining and an AI classifier and gathered information for 507 patients. For their examination, they utilized different AI approaches for training purposes utilizing ANN, and the SGD calculation gave them the best worth, which was 95%.

Amini et al. [61] led exploration to anticipate stroke rate, gathered 807 healthy and prone subjects in their investigation classified 50 danger factors for stroke, diabetes, cardiovascular illness, smoking, hyperlipidemia, and liquor use. They utilized two procedures that had the best exactness from c4.5 choice tree calculation, and it was 95%, and for K-closest neighbor, the precision was 94%.

Cheng et al. [62] produced a report on the assessment of the ischemic stroke risk. In their investigation, 82 ischemic stroke patient information were utilized, two ANN models were utilized to discover precision, and 79% and 95% were utilized.

Cheon et al. [63] conducted a research to foresee stroke patient mortality. In their investigation, they utilized 15099 patients to distinguish stroke event. They utilized a deep neural organization way to deal with recognize strokes. They utilized PCA to remove clinical record history and foresee stroke. They have 83% area under the curve (AUC).

Singh et al. [64] performed a study on stroke likelihood prediction using AI. In their exploration, they utilized an alternate strategy for foreseeing stroke on the cardiovascular health study (CHS) dataset. They have used principal component analysis for feature extraction and used decision tree algorithm. They utilized a neural net order calculation to build the model they got 97% accuracy.

Chin et al. [65] played out an examination to identify an early ischemic stroke. In their examination, the fundamental design was to foster a framework utilizing CNN to predict ischemic stroke. They gathered 256 pictures to prepare and test the CNN model. In their framework image preprocessing eliminate the impossible region that is not related to stroke, they utilized the data prolongation technique to raise the gathered picture. Their CNN technique has given 90% accuracy.

Sung et al. [69] performed a research to develop a stroke severity index. They gathered 3577 patient's information with acute ischemic stroke. For their predicting models, they utilized different data mining strategies and linear regression. Their best outcome was achieved from the k-nearest neighbor model (95% CI).

Monteiro et al. [66] performed a research to get a functional result prediction of ischemic stroke utilizing AI. In their exploration, they apply this method to patient who have spend three months after they got admitted. They got the AUC result above 90%.

Kansadub et al. [67] performed a study to anticipate stroke risk. In the investigation, the creators utilized Naive Bayes, Decision Tree, and Neural Network to break down information to foresee stroke. In their investigation, they utilized precision and AUC as their pointer's assessment. They have achieved best results using Decision tree algorithm.

Adam et al. [68] performed a research to categorize ischemic stroke. They utilized two models: a k-nearest neighbor and a decision tree algorithm to characterized ischemic stroke. In their examination, the decision tree algorithm was more usable for clinical experts who utilized it to group and identify stroke.

**2.3 PROPOSED SYSTEM**

In this project, we’ll be going through a five-step process for creating our project:

1. Evaluating data from “Stroke Prediction Dataset” using Matplotlib, seaborn & Plotly libraries.

2. Comparing different machine learning classifiers on the basis of accuracy on our dataset.

3. Performing k-Fold Cross-Validation and Hyper Parameter optimization on the model with highest accuracy. Also, identifying the best features for the model.

4. Preparing a backend model using Flask and the using the best saved model for prediction.

5. Getting input from the user using web app hosted on Heroku and predicting chances of stroke to him/her.

**2.4 OUR APPROACH**

In this project we have compared 10 different machine learning classifiers namely: Gaussian Naïve Bayes [3][4], Bernoulli Naïve Bayes [5][6][7], Logistic Regression [8][10][11][12][13], Decision Tree [14][15][16][17], Random Forest [25][26][27][28][29], K Nearest Neighbours [9], Stochastic Gradient Decent, Gradient Boosting [30], Support Vector Machine [18][19][20][21] and Neural Nets on “Stroke Prediction Dataset” and compared them based on their accuracy on the dataset. Our dataset consists of 10 attributes which are key factors leading to stroke which are: age, gender, hypertension, heart disease, marital status, work type, residence type, average glucose level, BMI and smoking habits.[1] We have achieved highest accuracy from K Nearest Neighbours of 96.23% on our dataset. More about the construction, working, implementation and results is covered in upcoming chapters.

**CHAPTER – 3**

**PROBLEM DESCRIPTION AND SPECIFICATION**

**3.1 OVERVIEW**

In this chapter we are going to discuss about the problem statement and specification that we wish to offer through this project. The chapter highlights our approach and defines the functionalities that we wish to deliver through this project. We will look after the process flow diagram of our project and also get to the requirements for carrying out this project. This chapter lays the foundation of our work which is covered in later sections.

**3.2 PROBLEM DESCRIPTION**

To compare various machine learning algorithms based on their accuracy of stroke prediction and build a system for the same.

**3.3 SPECIFICATION**

Stroke prediction is a task that involves machine learning concepts to recognize the possible factors contributing to the conditions developing strokes and to help prevent them using analytics to analyses the possible scenarios to stop or prevent it.

The aim of project is to apply various machine learning algorithms on the “Stroke Prediction Dataset and evaluate the models based on their accuracy. Also, create an interface where a user can provide his medical details and the prediction model will provide the chances of the stroke to him. Moreover, a comparison between various machine learning algorithms helps us in determining the most suitable algorithm for the prediction model. This prediction model can aid in clinical decision making and help patients to have an improved and reliable stroke risk prediction. In future, we can use more risk factors to train our model and enhance the accuracy of the model. The web interface will be friendly and also, we can provide valuable suggestions to users as per their stroke risk.

Figure 3.1 describes the process flow diagram for the approach we used in this project. We performed different evaluation and analysis on our dataset to know different mapping of attributes. We analysed the percentage of various values in different attributes, how they contribute to the model, we detected outliers and anomalies present in the dataset and removed them. We also handled the missing values. Then we split our dataset for training and testing purpose and used ten different machine learning classifiers and compared them based on their accuracy. We also calculated precision, F1 score, recall, specificity for all the models. Then we used the best model and performed k-fold validation and hyper-parameter optimization on it. We saved this model and used it for predicting the likelihood of stroke to the user whose data we collected at real time through our online hosted web app.

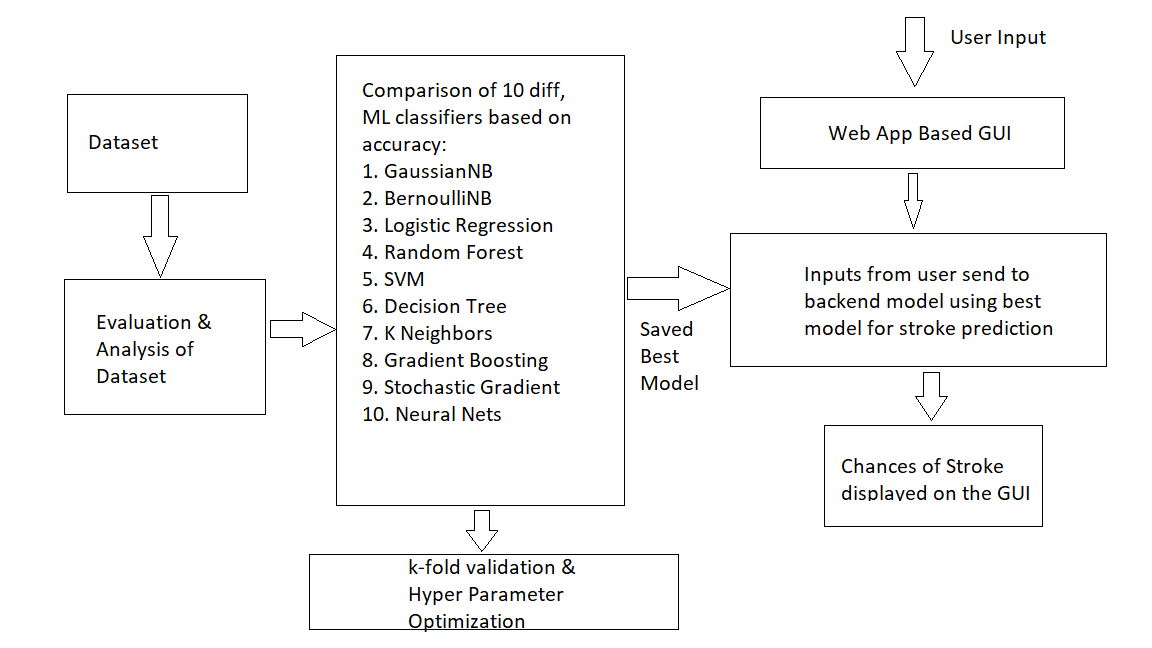


Figure 3.1: Process Flow Diagram

**3.4 REQUIREMENTS**

**3.4.1 SOFTWARE REQUIREMENTS**

* Python Version 3.6+
* IDE: JupyterLab or PyCharm
* Frameworks: -
  + Flask
  + Bootstrap
* Python Libraries: -
  + Scikit-learn
  + Seaborn
  + NumPy: 1.18.5
  + Pandas: 1.1.14

**3.4.2 HARDWARE REQUIREMENTS**

* Processor: - Intel i3+ or AMD A6+
* RAM: - 2GB/+
* Hard Disk Space: - 2GB

**3.4.3 OTHER REQUIREMENTS**

* Stroke Prediction Dataset
* For training the model - Kaggle Notebook/Google Colab/JupyterLab should be preferred (Recommended)

**CHAPTER – 4**

**SYSTEM DESIGN**

**4.1 INTRODUCTION**

According to World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of the total deaths [70]. Through this project we will compare various machine learning algorithms based on their accuracy of stroke prediction and build the system for the same.

The model will consist of a front-end GUI for collecting data. Flask file to connect the model to backend model. For the backend model we will be comparing 10 different ML algorithms for their accuracy. Select the best one and perform K-fold Cross validation [33][34][35][36][37] and hyper-parameter optimization [38][39][40][41][42] on it.

**4.1.1. Data Sources**

We had selected “Stroke-prediction dataset” obtained from Kaggle as our dataset[1]. This dataset contains of 5110 records having 12 attributes (id, age, gender, hypertension, heart disease, marital status, residence type, work type, average glucose level, BMI, smoking status and stroke occurrence).

Figure 4.1 shows the different attributes in our dataset along with their count, nature (null or not null) and their data type (binary or categorical). There are three attributes of integer and float type and five attributes of object type. We can also notice that BMI attribute has some missing values. In the upcoming sections we will handle these issues.

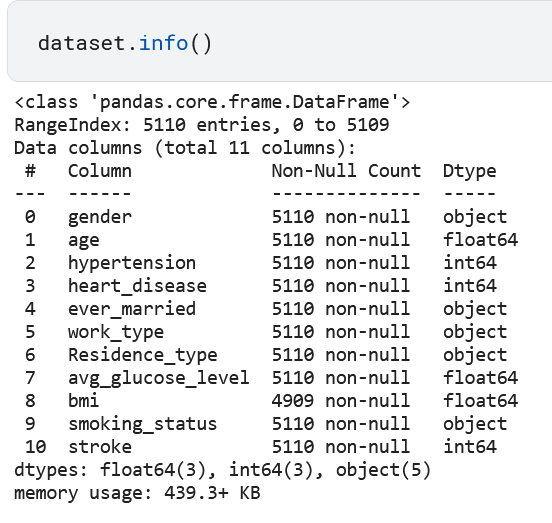


Figure 4.1: Output showing the dataset information

**4.1.2. MACHINE LEARNING ALGORITHMS USED IN THE PROJECT**

In this project we are going to train our dataset on 10 different machine learning algorithms and based on their performance we will select the best model and further use it in building the stroke prediction system. The 10 machine learning algorithms used in this project are: -

1. Gaussian Naive Bayes
2. Bernoulli naïve Bayes
3. Logistic Regression
4. Support Vector Machines
5. Random Forest
6. Decision Tree classifier
7. K-Nearest Neighbor
8. Gradient Boosting
9. Stochastic Gradient Descent
10. Neural nets

**4.1.3. EVALUATION METRICS**

Model parameters are the variables that helps us in evaluating the model. Their values can be estimated on the basis of the dataset used in the model. The various parameters we computed for the machine learning algorithms we covered are: -

* 1. **Accuracy**- It is the ratio of number of correct predications to the total number of input samples.[71]

The formula for accuracy obtained from confusion metrics is:

(i)

* 1. **Precision**- Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances.[72]

(ii)

Where, TP is True positives

FP is False Negatives

* 1. **Recall**- Recall (also known as sensitivity) is the fraction of relevant instances that were retrieved.[72]

(iii)

Where, TP is True Positives

FN is False Negatives

* 1. **F1 Score**- In statistical analysis of binary classification, the F-score or F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test.[73]

The formula for F1 Score is-

(iv)

* 1. **Specificity**- Specificity (True Negative rate) measures the proportion of negatives that are correctly identified (i.e., the proportion of those who do not have the condition (unaffected) who are correctly identified as not having the condition).[74]

Mathematically specificity is given by-

(v)

**4.2 ARCHITECTURE DIAGRAM**

The stroke predicition system using machine learning predicts the risk of stroke to the user based on the various symptom information the user gives, such as age, gender, marital status, hypertension, prior heart disease and smoking status. The architecture of the stroke risk prediction using machine learning consists of the dataset through which we will compare the symptoms of the user. Model is trained using the dataset. The classification algorithms process the data and predicts the risk of stroke. The diagram below explains about the system in perception of overview of the system.

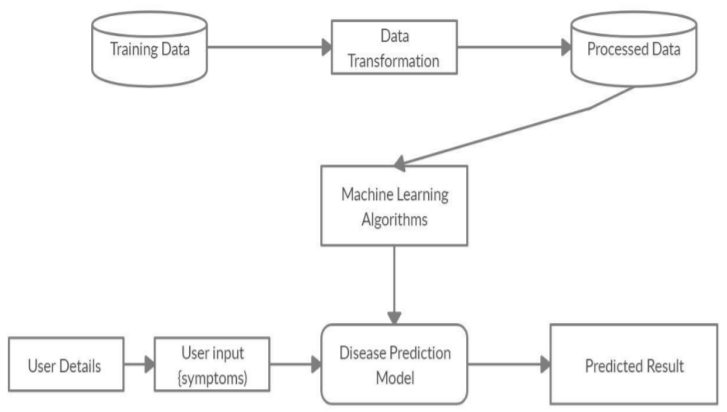
****

Figure 4.2: Architecture of the Stroke prediction model

**4.3 SYSTEM DESIGN**

**4.3.1 USE CASE DIAGRAM**

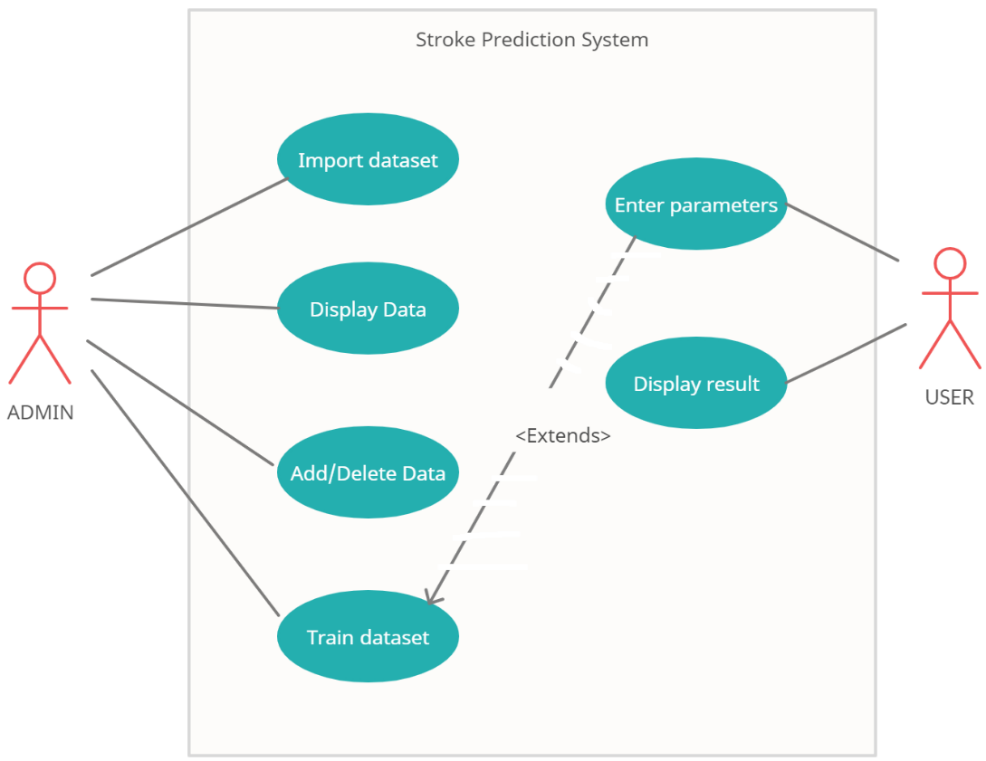
****

Figure 4.3: Use Case Diagram for stroke prediction system

Figure 4.3 shows the Use Case Diagram of the project. It shows the various association of user and admin with the stroke prediction system. The admin is responsible for the training and enhancing the model whereas, the user interacts with the model by entering the parameters of his medical condition and getting the results as predicted by the model.

**4.3.2 DATA FLOW DIAGRAM**

Data flow diagram focuses on flow of data through the program. It shows where the inputs are received from, where it is processed, where it is stored and where from the output is given. It consists of four basic diagrams: A circle or square with rounded edges to represent a process. A square to represent an external entity. A pair of parallel lines or a box with D mentioned in it to represent a data store. And arrows (unidirectional and bidirectional) to represent the flow of data through the system. The entire working or the flow of the data can be divided into three groups for better understanding. They are- 1. DFD-L0 2. DFD-L1 3. DFD-L2. Each containing different level of details in the view of processes to make it easier to follow.

**4.3.2.1 Data Flow Diagram Level 0**

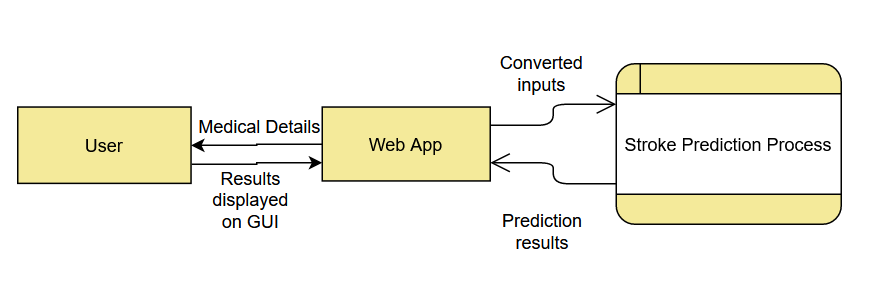
****

Figure 4.4: DFD level 0

Figure 4.4 shows DFD level 0 for our project. It consists of an external entity i.e., user represented in box which passes it medical details as data to another external entity known as Web App which again passes this data in converted form to Stroke prediction process represented in square box with rounded edges. The process generates the result and send back the data to Web App which displays the output to the user based on the values from the Stroke prediction process.

**4.3.2.2 Data Flow Diagram Level 1**

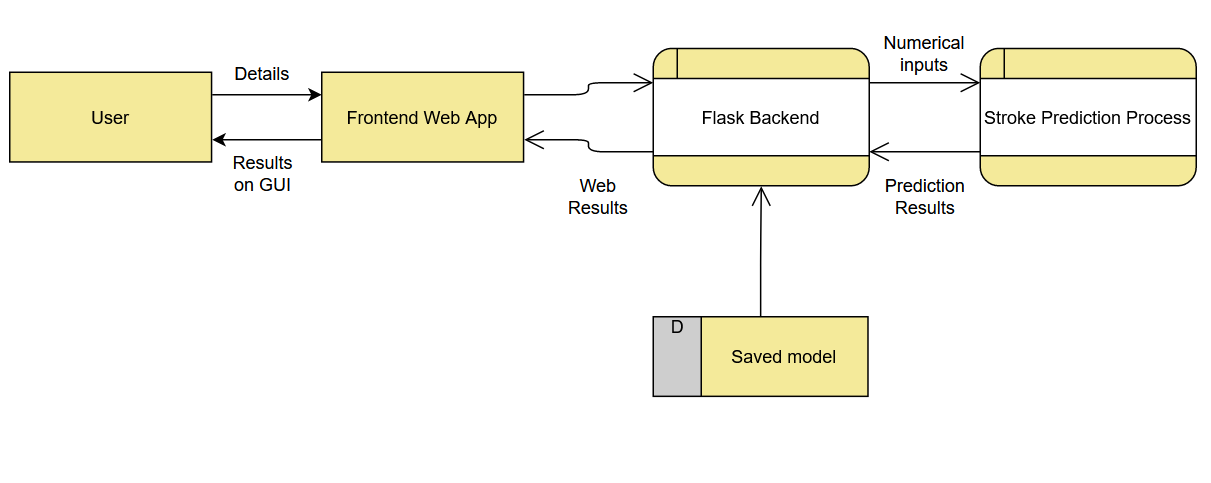
****

Figure 4.5: DFD level 1

Figure 4.5 represents DFD level 1. It shows how the front-end process consists of a backend process which receives it data from the front-end web app and uses the saved best model. It passes these inputs to the stroke prediction process which uses the saved model and the input parameters and predicts the output and returns the result back to flask-based backend process. The backend process then selects the web output results to be displayed to the user and passes it to the external entity called web app. User is able to see the results through this external entity.

**4.3.2.3 Data Flow Diagram Level 2**

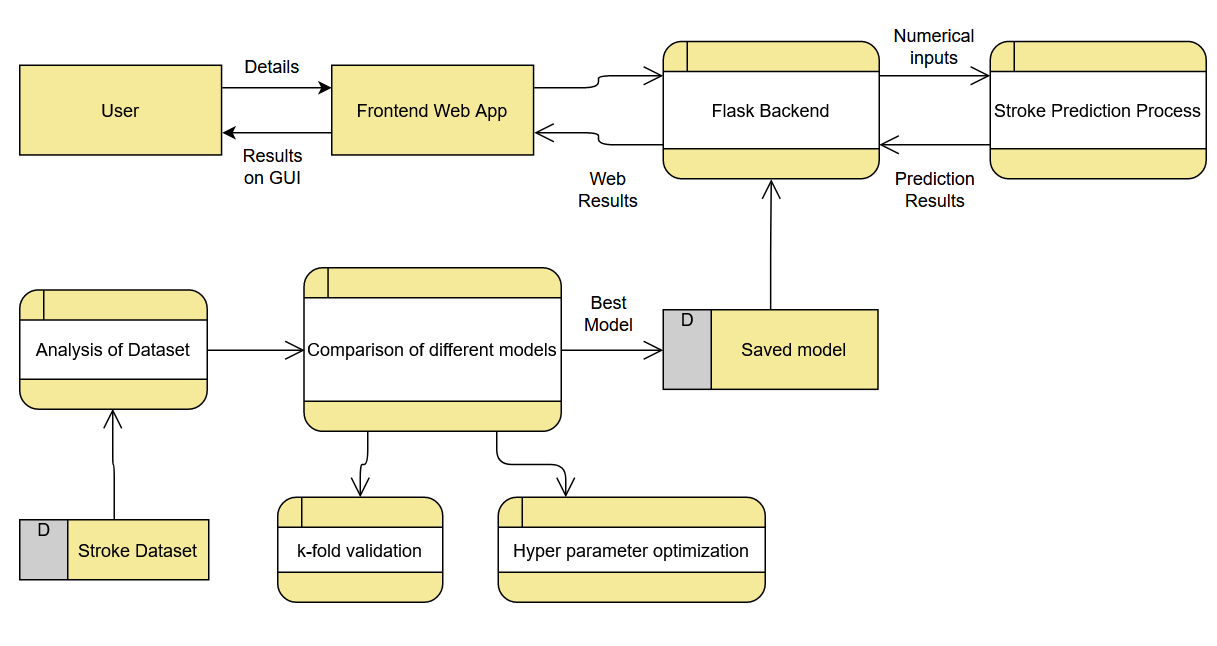
****

Figure 4.6: DFD level 2

Figure 4.6 represents level 2 data flow diagram. It shows a detailed view of process involved in generation of saved model. Initially a data store containing Stroke Prediction Dataset is stored in the data store. This is passed to the Analysis process for the purpose of analysis, which further contains of process like anomaly detection, handling of missing values, handling of outliers etc. and once the train test split is prepared using this process, it is passed to a process known as Comparison of different models. In this process ten different machine learning classifiers are used to train on this dataset and accuracy is tested. The best obtained model is saved in the Data store to be used by Backend flask process. Once the input is received by the user and given to backend process, it uses this saved model and passes it along with converted numerical values to the stroke prediction process. The stroke prediction process calculates the results and return it to flask backend process. This process based on the results from stroke prediction process, selects the output to be displayed and passes it to the external entity known as frontend which displays the result to the user.

**4.4 SEQUENCE DIAGRAM**

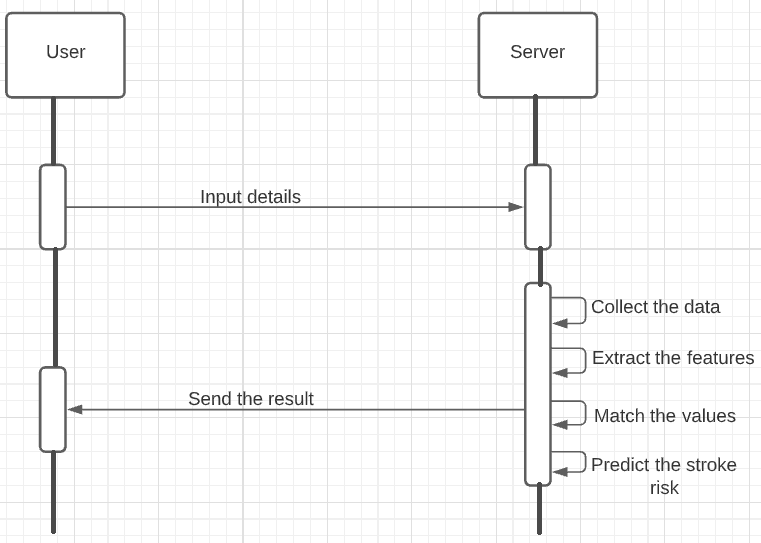
****

Figure 4.7: Sequence Diagram of the system

Sequence diagrams shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario [71]. In our project it follows a very straight forward approach; the user visiting the web app inputs the values, which are passed to the server where the prediction model predicts the results which are then again sent back to the user on the web app.

**4.5 FLOW CHART FOR STROKE PREDICTION PROCESS**

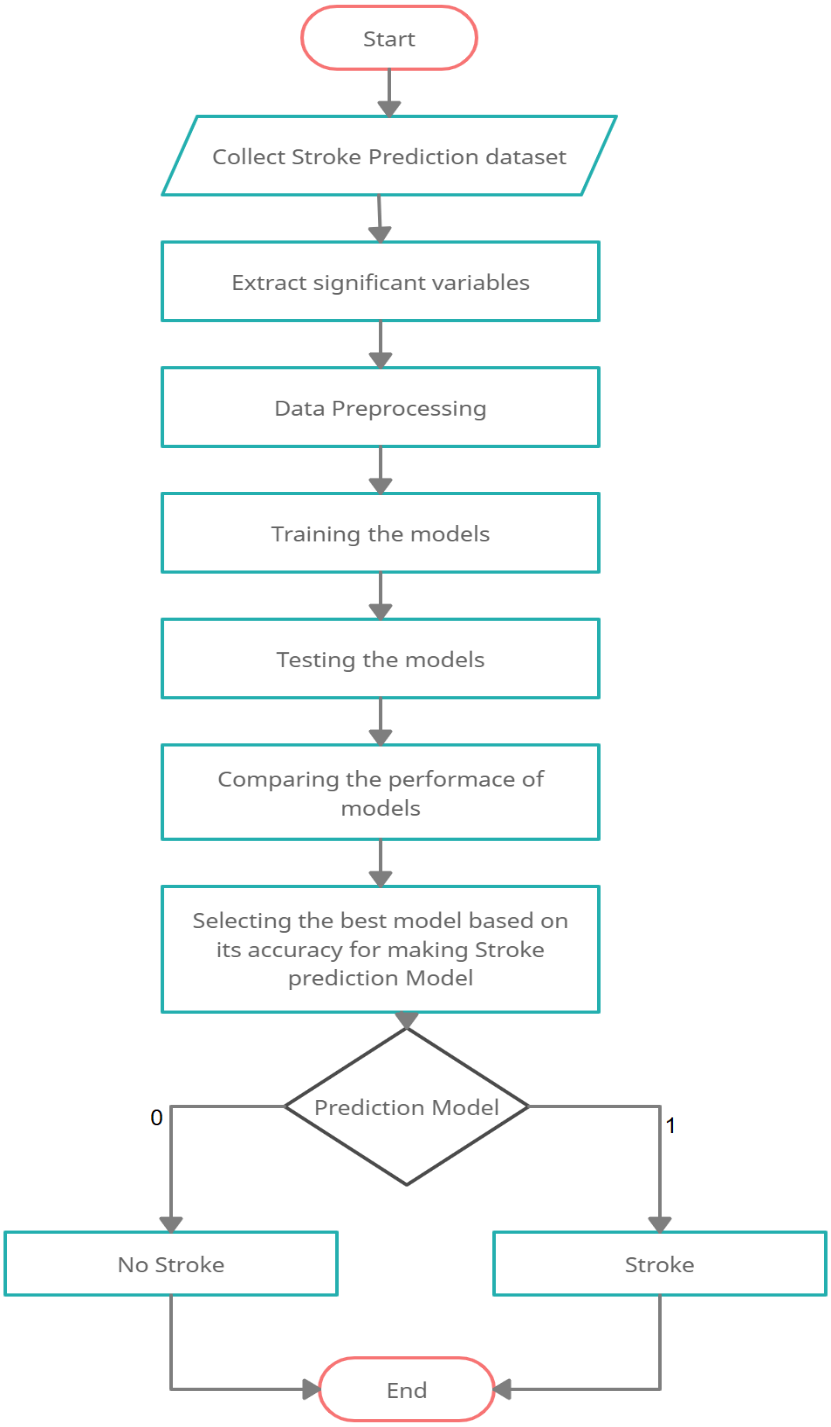
****

Figure 4.8: Navigation flow for the stroke prediction system

Figure 4.8 shows the overall flow of work in the stroke prediction system. Initially user data is collected using web-based GUI platform. The data fetched from the web app is passed to prediction model in numerical form via flask backend model. The model processes the data and return 0 for no risk and 1 for risk of stroke which again is given to flask model which then displays the result based on values from the prediction model to the web app. This result is displayed to the user on a web-page.

**CHAPTER – 5**

**IMPLEMENTATION AND RESULT**

**5.1 OVERVIEW**

In this section, we are going to cover all the steps we have taken while implementing the program. The whole program consists of different sections giving a brief about the program implementation and their respective outputs.

**5.2 IMPORTING THE NECESSARY LIBRARIES AND DATASET**

We have used a number of python libraries in our project. Below is a snap of some of the main libraries used in the project.

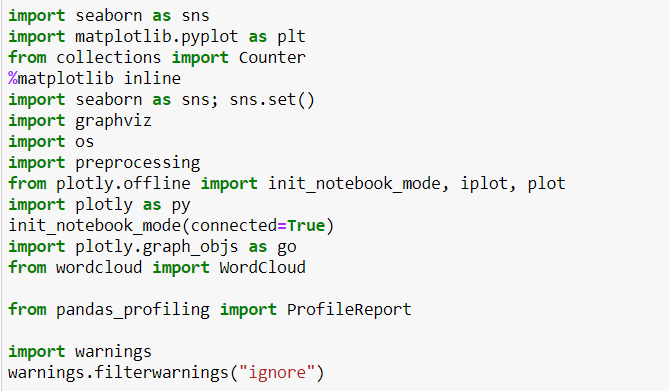


Figure 5.1: Code snippet showing the import of libraries used in the project



Figure 5.2: Code snippet to read the dataset

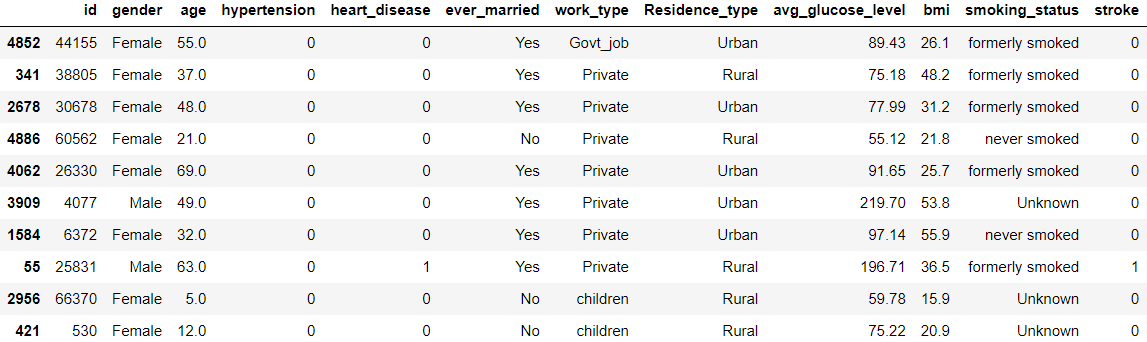


Figure 5.3: 10 Sample data from the dataset

We’ll drop the ‘id’ column as it may cause unwanted correlation.

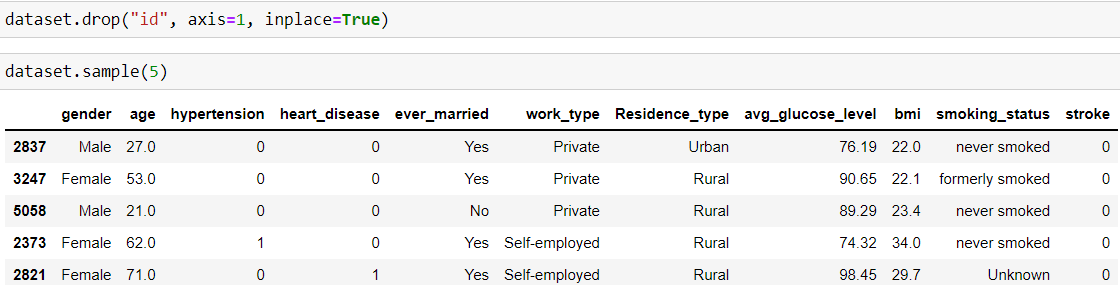


Figure 5.4: - Code snippet and resultant table after dropping the id column

**5.3 BASIC DATA ANALYSIS**

In this section we used matplotlib, seaborn and plotly libraries to evaluate and analyze relationship among different attributes and their contribution in stroke. First, we analyzed the constituent of various values in different attributes to analyze which of the value is more occurring in the case of stroke risk.

Figure 5.4a represents a distribution of records based on their smoking habits as present in our dataset. Similarly, we have analyzed the percentage for other attributes like gender, work type, residence location etc. Figure 5.4b shows a box plot representing the age and stroke, it basically shows persons from which age have suffered more from stroke. The darker the color the greater number of cases of stroke belong to that range. Similarly, the relationship among body mass index and stroke shows that which BMI level individuals has the highest occurrence of stroke. The next one shows the distribution of records of average glucose level and its relationship with stroke case. The darker region shows that persons with average glucose level in that range had suffered more from stroke.

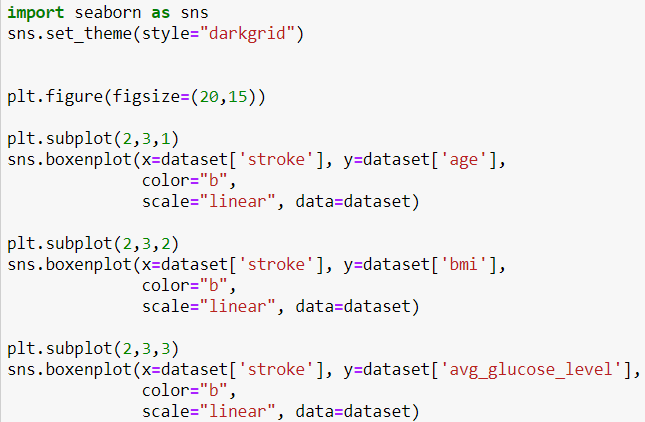




Figure 5.5: Code snippet to visualize the dataset

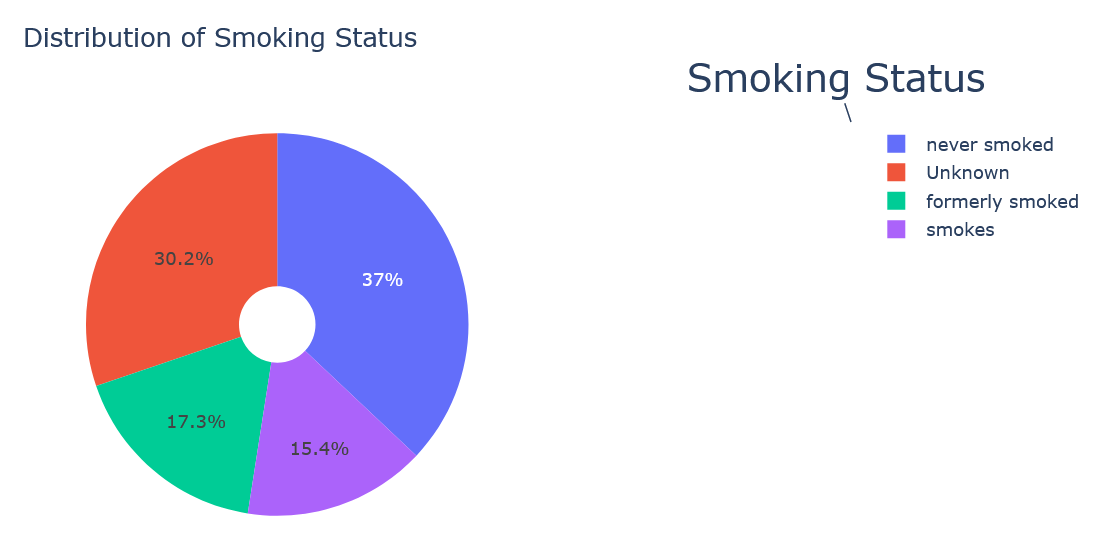


Figure 5.5a. Pie chart representing the distribution of smoking status of individuals in percentage in the dataset

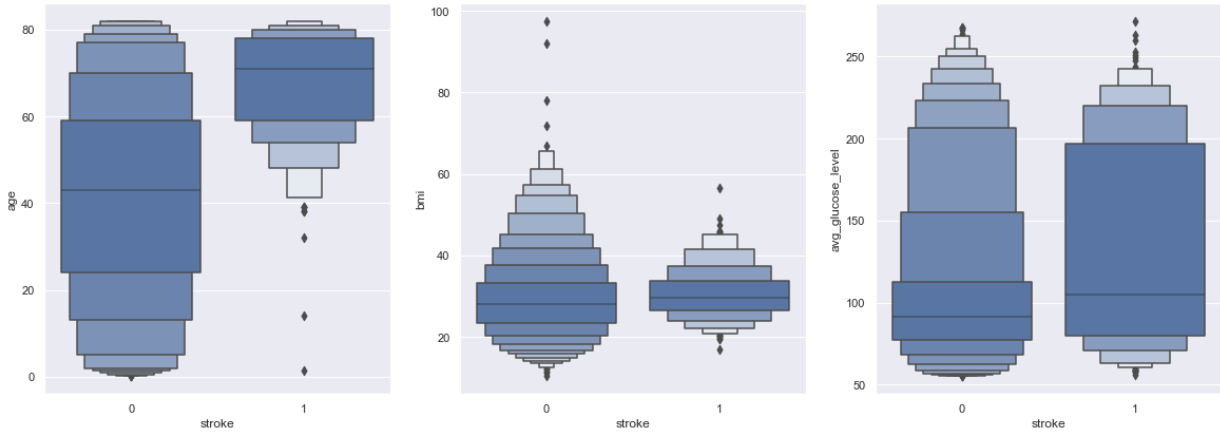


Figure 5.5b. Boxenplot on the dataset showing relation of stroke with age, BMI and average glucose level

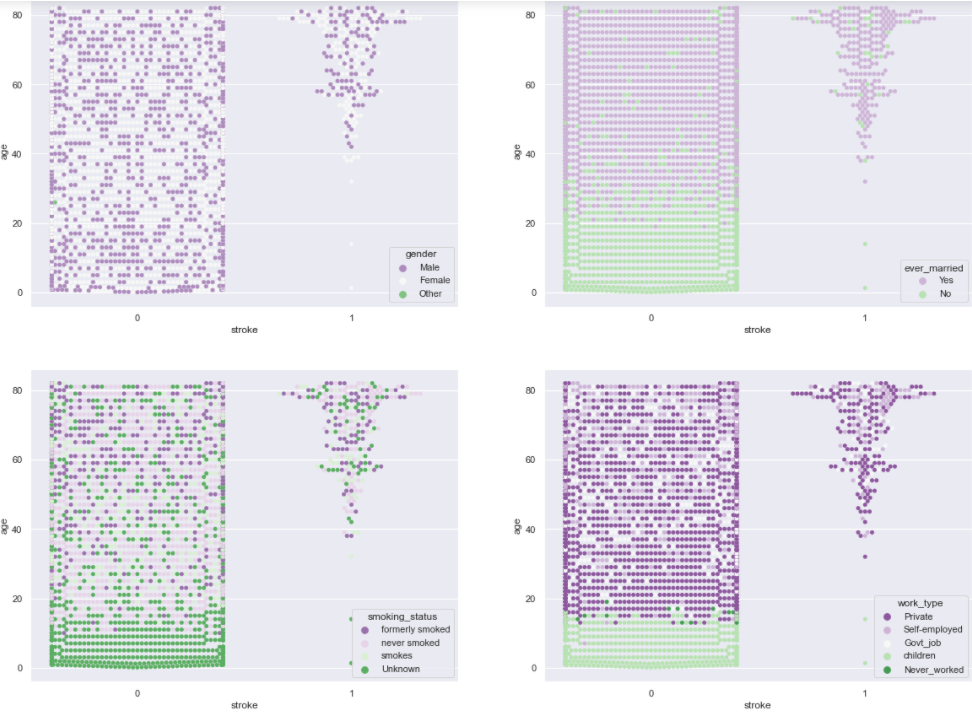


Figure 5.5c. Figure representing swarm of individual records representing relation between age, gender, marital status, smoking habits, work type and likelihood of stroke

Figure 5.4c top left corner sub section represents how people from different age group had suffered from stroke belonging to different gender classes. On the x axis 0 signify no stroke history and 1 signify that person with that record has suffered from stroke. The top right corner image represents how marital status is a contributing factor in stroke risk. The purple represent that the person is married and green represents that he/she is not married. The bottom left image shows the relationship between age, smoking status and stroke. It signifies how person belonging to different age group having varying smoking habits had suffered from stroke. It shows how smoking contributes in stroke risk. The bottom right image signifies the relationship among work type and stroke risk. It signifies that work type is also a factor in stroke risk and how people belonging to different work group had suffered stroke.

**5.4 PANDAS PROFILING**

Pandas profiling is a useful library that generates interactive reports about the data. With using this library, we can see types of data, distribution of data and various statistical information. This tool has many features for data preparing. Pandas Profiling includes graphics about specific feature and correlation maps too. We have used this to detect the distribution, correlation and analyze the minimum, maximum and mean values of different attributes. It also helped us to detect the missing values and the outliers in different attributes.



Figure 5.6: Code snippet showing pandas profiling

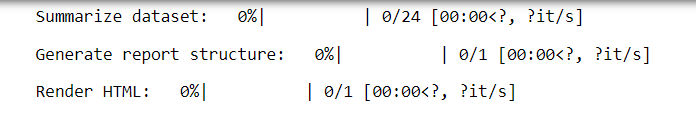


Figure 5.6a: Pandas profile report

**5.5 CORRELATION**

Correlation explains how one or more variables are related to each other. These variables can be input data features which have been used to forecast our target variable. **Correlation**, is a technique which determines how one variables moves/changes in relation with the other variable. If two variables are closely correlated, then we can predict one variable from the other. Correlation plays a vital role in locating the important variables on which other variables depend. **Positive correlation** means when one variable increases the other also increases. **Negative correlation** means when one variable increases other decreases. And in case of no correlation the increase or decrease in one variable does not affect other variable.

Figure 5.6 represents the heatmap of correlation between different attributes in the dataset. A heatmap is a graphical representation of correlation between attributes using colors to represent the extent of correlation. The darker the color (black or grey) the lower the correlation and the brighter the color (red or pink) the higher the correlation. In the figure the value of correlation among these attributes are also mentioned in the cell present at the intersection of attributes. This figure tells us how much different attribute depends on the other attribute. This also leads to identification of significant attribute which plays a crucial role in development of relation with other attribute and in turn helps in deciding the result of prediction of stroke.

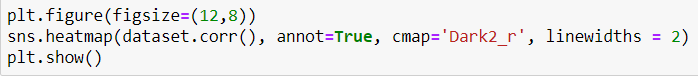


Figure 5.7: Code snippet to show correlation

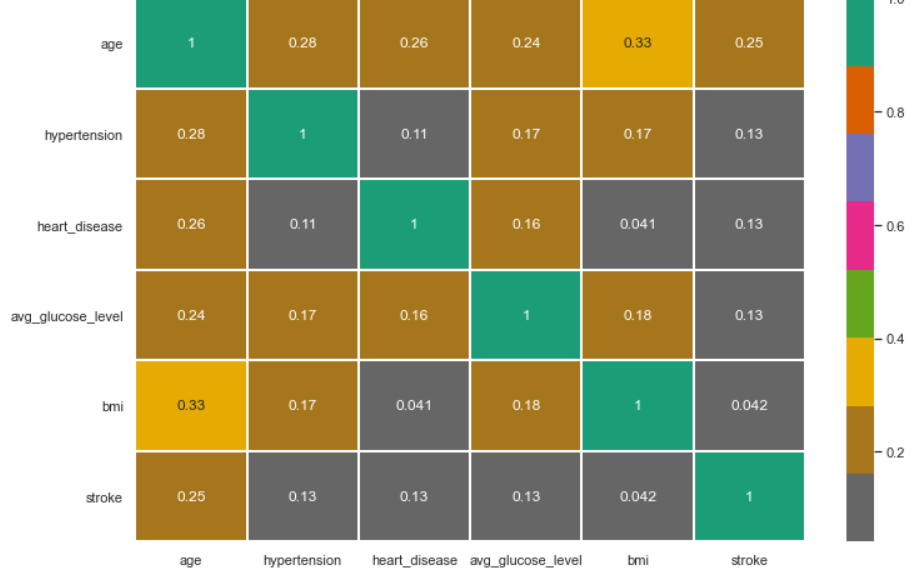


Figure 5.8: Sample output showing the heatmap of correlation

Figure 5.6 shows that age and BMI have the highest correlation of 0.33. It also shows that hypertension and heart disease has significantly less correlation as per the dataset. Along with that all the grey values signify that those attributes hold very less correlation and does not affect each other to a determining level. The diagonal values are 1 because every attribute holds a perfect correlation with itself.



Figure 5.9: Code snippet for pairplot representing how different attributes is related to each other

A pairplot plot a pairwise relationships in a dataset. The pairplot function creates a grid of axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column. **Pairplot** uses to get the relation between each and every variable present in [Pandas](https://indianaiproduction.com/pandas-dataframe/) DataFrame. It works like a seaborn scatter plot but it plots only two variables plot and seaborn pairplot plot the pairwise plot of multiple features/variables in a grid format. The value in the x axis increases and the corresponding relationship with each attribute is represented on y axis of that particular attribute. Figure 5.7 shows how varying an attribute (increasing value on x-axis) impacts the other attributes shown in same column (on their respective y axis). Stroke is selected as the target variable. For example, lets take age as the attribute on x – axis, then varying age from 0 – 100 and viewing its relationship with other attributes such as BMI for individuals spread across the range from 0 to 100 on y axis. Also, the blue color dot signifying no stroke history and orange dot representing a person with particular age and BMI level who has suffered from stroke. Similarly, this approach is used to understand how one variable effect other variable and together they contribute in stroke and also to what extent.

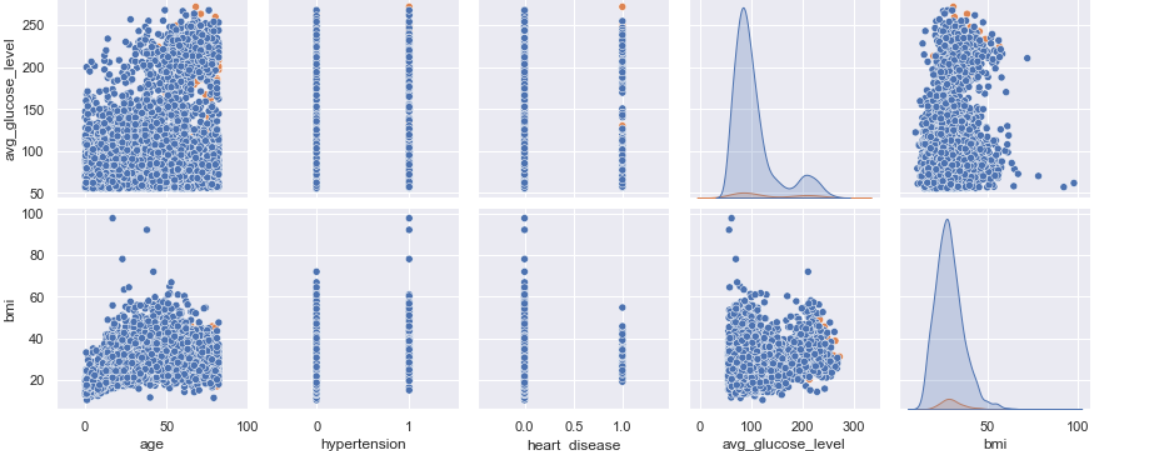
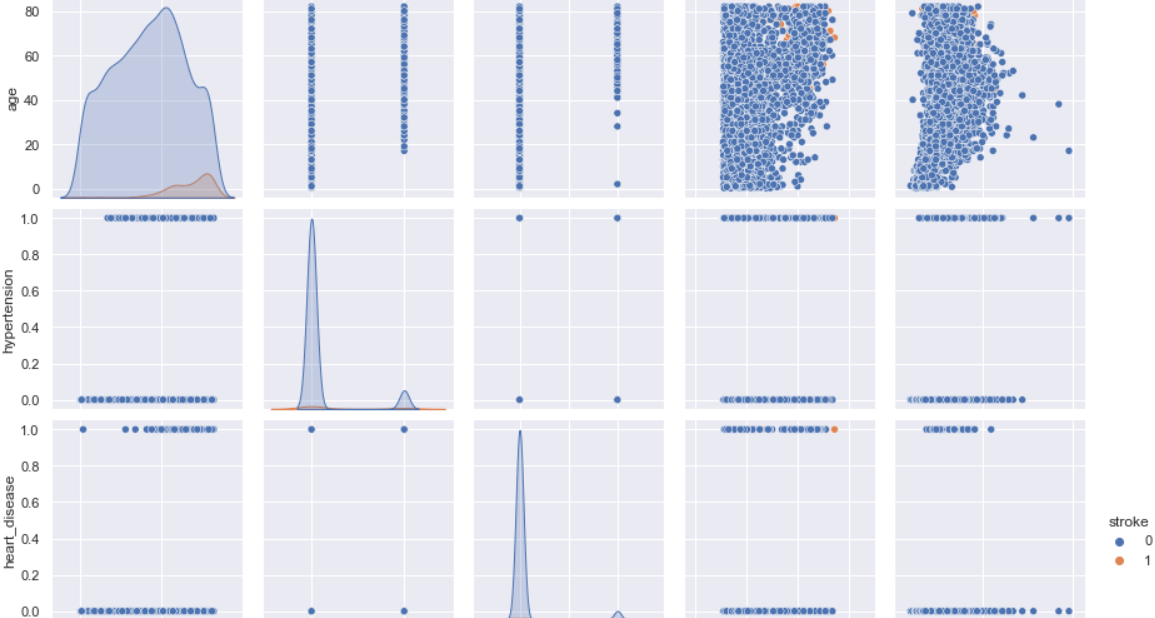


Figure 5.10: Sample output showing the distribution of values for each attribute in their domains, their relation with other attribute and how they play role in stroke risk.

**5.6 ANAMOLY DETECTION**

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

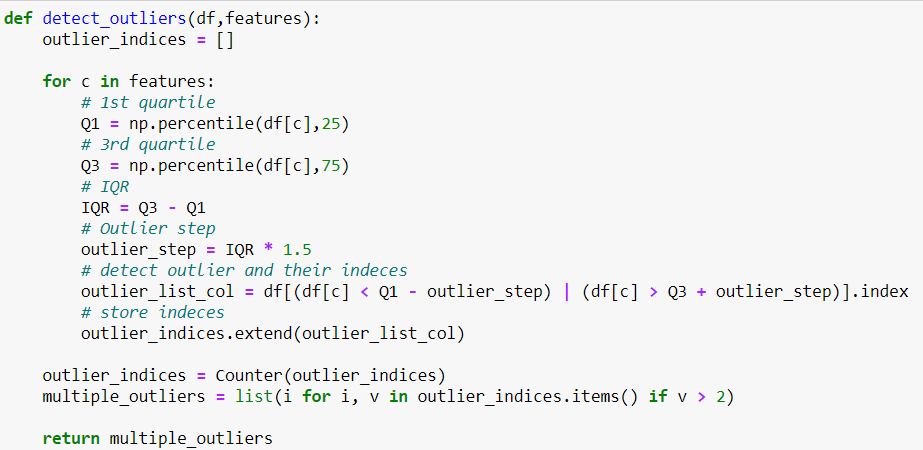


Figure 5.11: Code snippet showing the detection of outliers

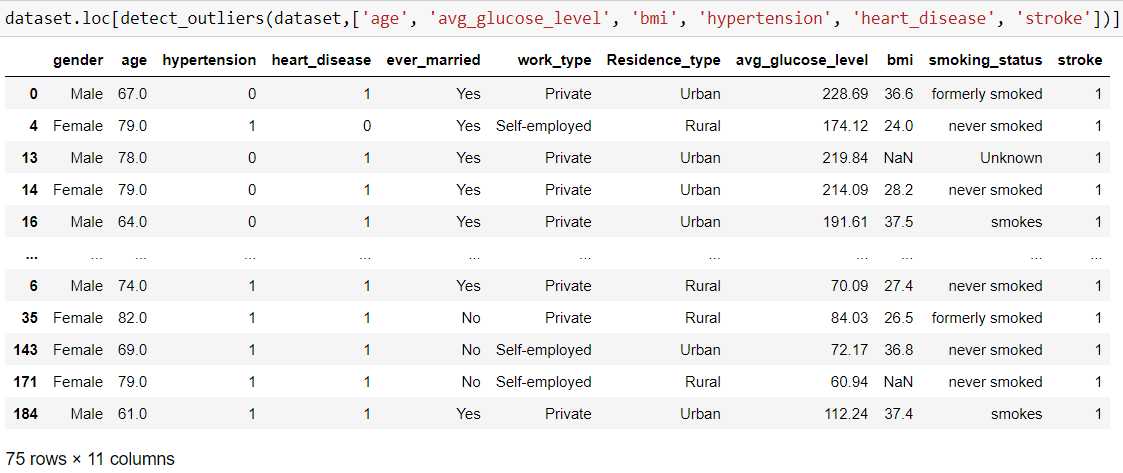


Figure 5.12: Output showing the detection of outliers

dataset = dataset.drop(detect\_outliers(dataset,['age', 'avg\_glucose\_level', 'bmi', 'hypertension', 'heart\_disease', 'stroke']),axis = 0).reset\_index(drop = True)

Figure 5.13: Code snippet to drop the dataset containing outliers

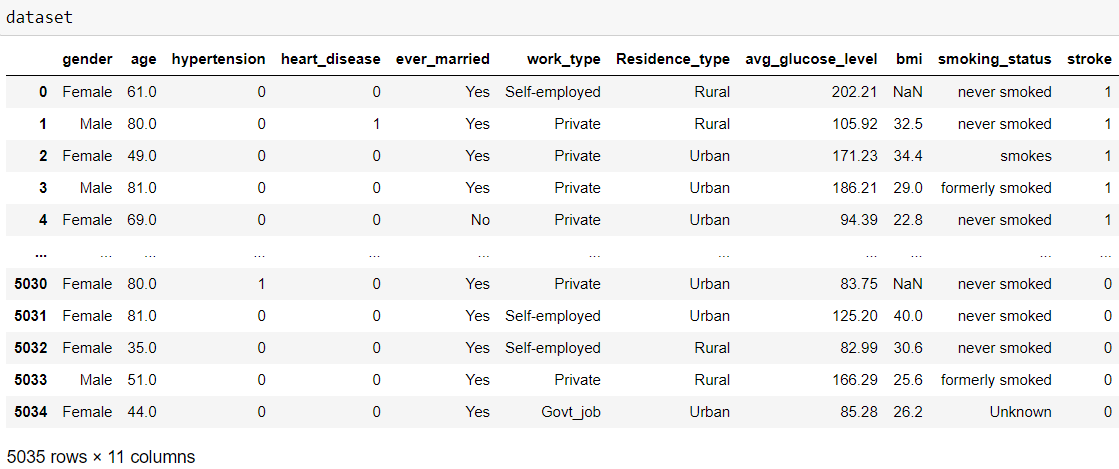


Figure 5.14: Resultant table after anomaly detection and rectification

**5.7 HANDLING MISSING VALUES**

We have 201 null values in total. After anomaly detection it reduced to 193.



Figure 15: Code snippet to check for missing bmi values.

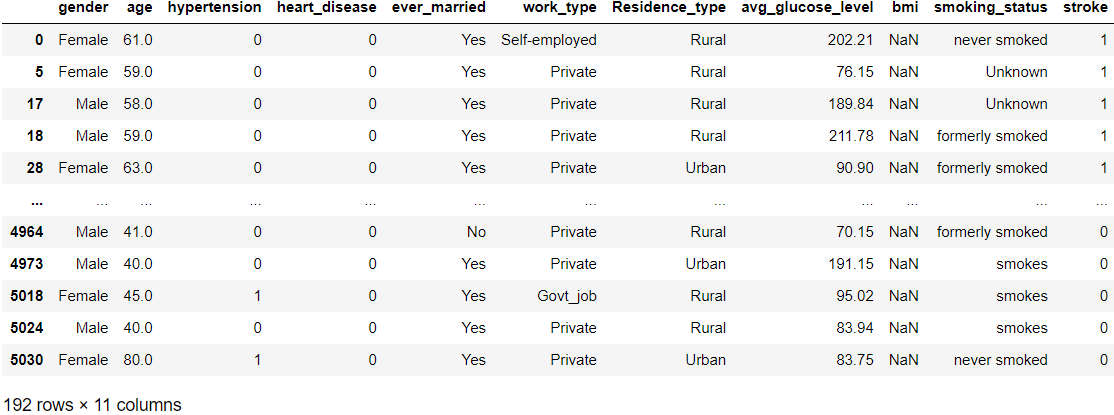


Figure 5.16: Output showing the rows where BMI value is null.

To handle missing values, we selected BMI based on different genders. We obtained the mean for male, female and other gender and then assigned this means to the missing values in the section of BMI against the respective gender. Figure 5.11 represents the box plot use to identify the minimum, maximum, mean value and outliers in different gender categories plotted on x axis with respect to their BMI level.

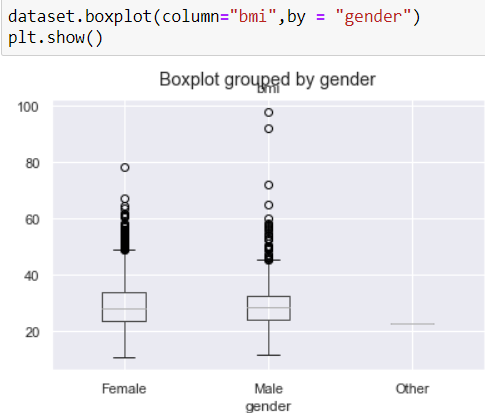


Figure 5.17: Output showing the boxplot grouped by gender

We get different BMI averages for women and men, although not very large. So, we assigned the total BMI mean as there are very few examples for the Others gender.

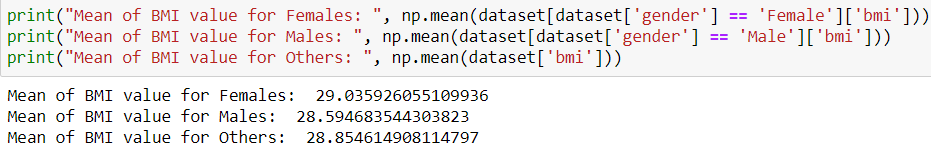


Figure 5.18: Code snippet and output showing the mean BMI based on gender

We fill the BMI field with null values to 0.



Figure 5.19: Code snippet and output showing that none of the field contains null or missing values

**5.8 ENCODING**

****

Figure 5.20: Code snippet for encoding

First, we will handle categorial values.

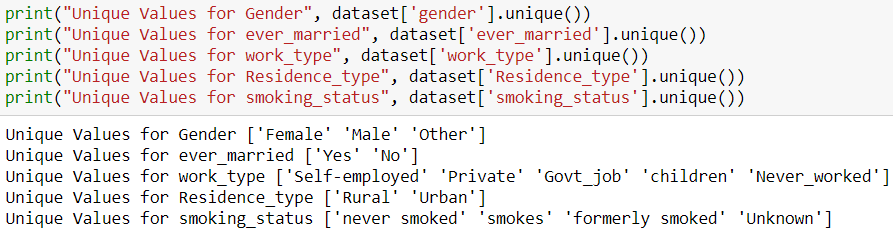


Figure 5.21: Code snippet and output for handling categorial values

**5.8.1 LABEL ENCODING**

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

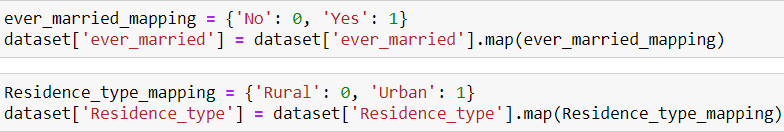


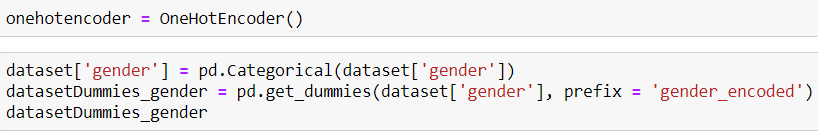
Figure 5.22: Code snippet to implement label encoding

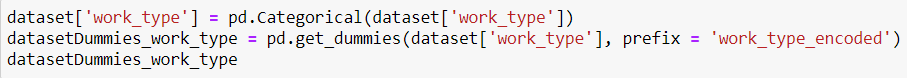
**5.8.2 ONE HOT ENCODING**

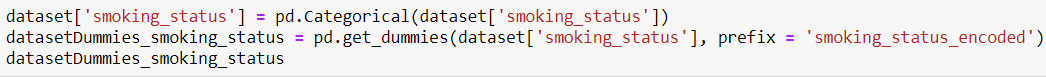
One Hot Encoding is the binary representation of categorical variables. This process requires categorical values to be mapped to integer values first. Next, each integer value is represented as a binary vector with all values zero except the integer index marked with 1.

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

Here, we converted categorical data to the binary values. This operation increases the accuracy.







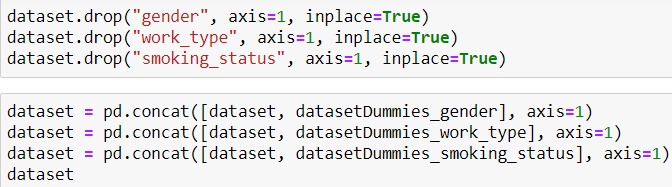


Figure 5.23: Code snippet for implementing one hot encoding

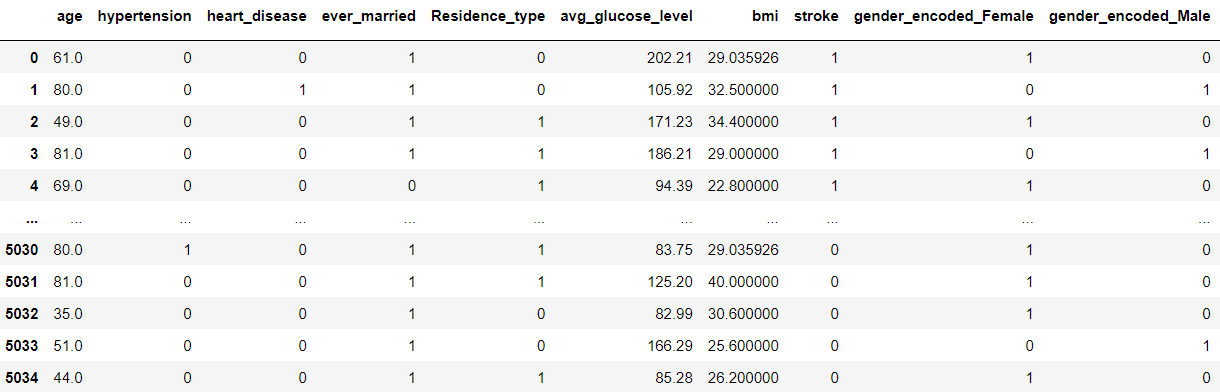


Figure 5.24: Sample output showing the table doesn’t contain categorial variables.

Dataset is now ready for implementing machine learning algorithms.

**5.9 SPLITING THE DATASET FOR TRAINING & TESTING**

Here we split the dataset in the ratio of 80:10:10 for train, validation and testing respectively. To maintain the variation in dataset we have selected the random samples using random attribute in the train\_test split function of the scikit learn.



Figure 5.25: Code snippet showing the dataset is split for the purpose of training, testing and validation.

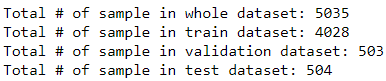


Figure 5.26: Output showing the amount of data used for training, testing and validation

**5.9.1 STANDARDIZATION**

Standardization is a method in which the mean value is 0 and the standard deviation is 1, and the distribution approaches the normal. The formula is as follows, we subtract the average value from the value we have, then divide it by the variance value.

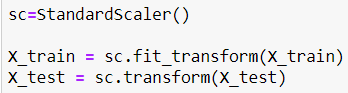


Figure 5.27: Code snippet to implement Standardization

**5.9.2 SAVING THE SCALER OBJECT**

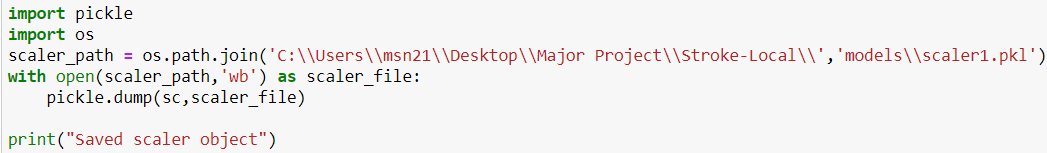


Figure 5.28: Code snippet for saving the scaler object

We have the model which has given us the highest accuracy for using it as the prediction model for the project.

**5.10 COMPUTING TRAINING SCORE OF MODELS**

The following are the ML algorithms that will apply to dataset. Results will contain train-validation-test scores, confusion matrix, statistical information and classification reports for each algorithm.



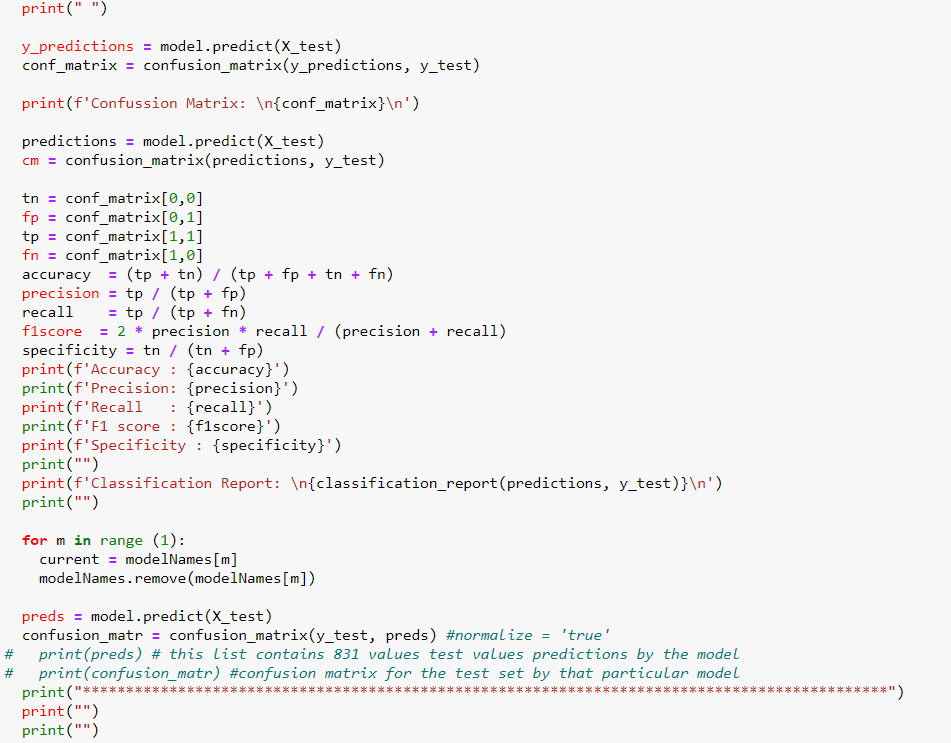
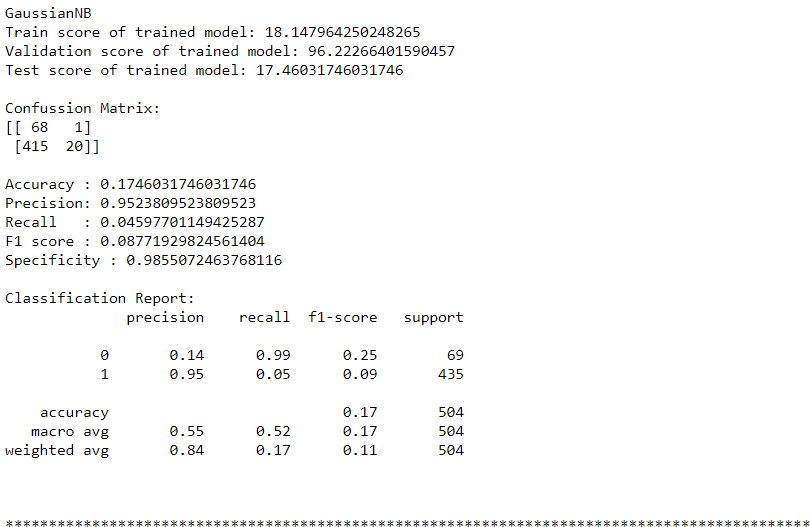
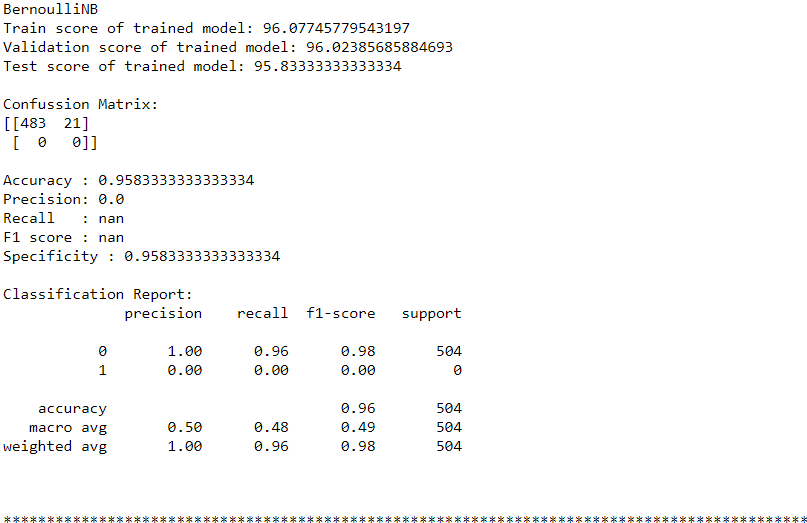
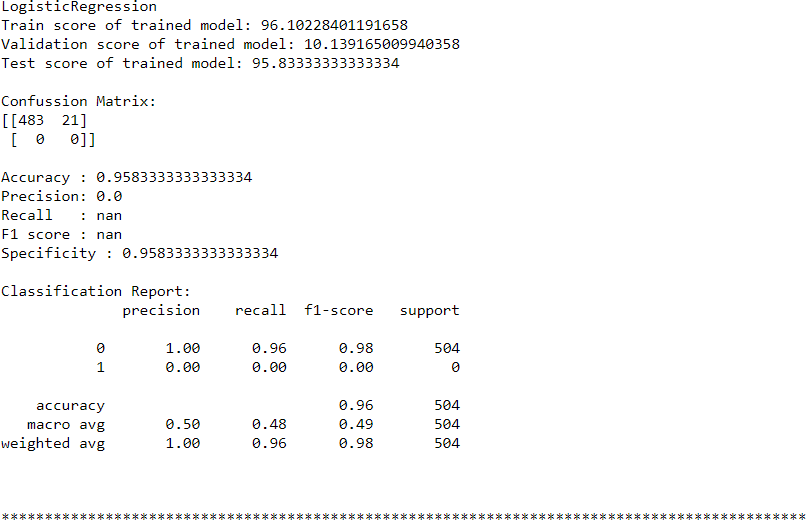
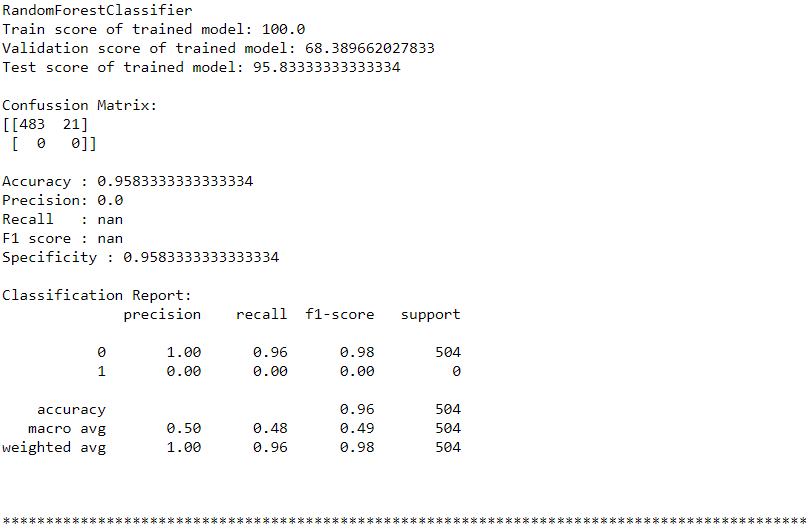


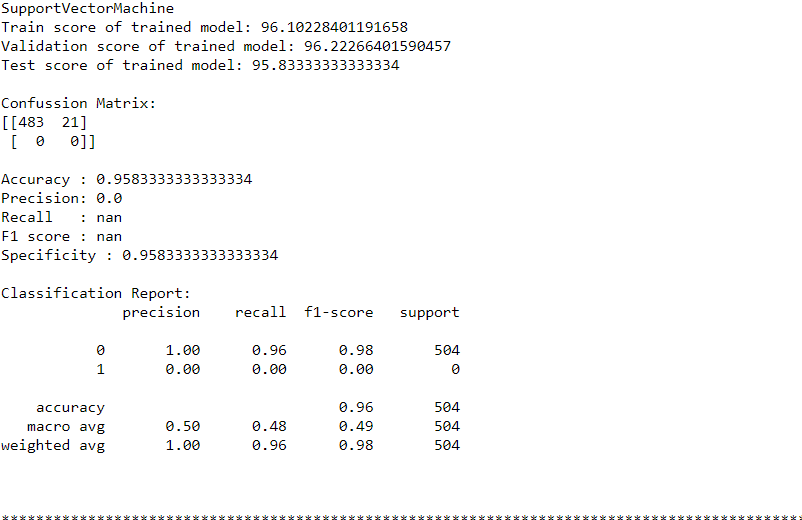
Figure 5.29: Code snippet for computing accuracy, precision, recall, F1 score, specificity and confusion matrix of various ML algorithms

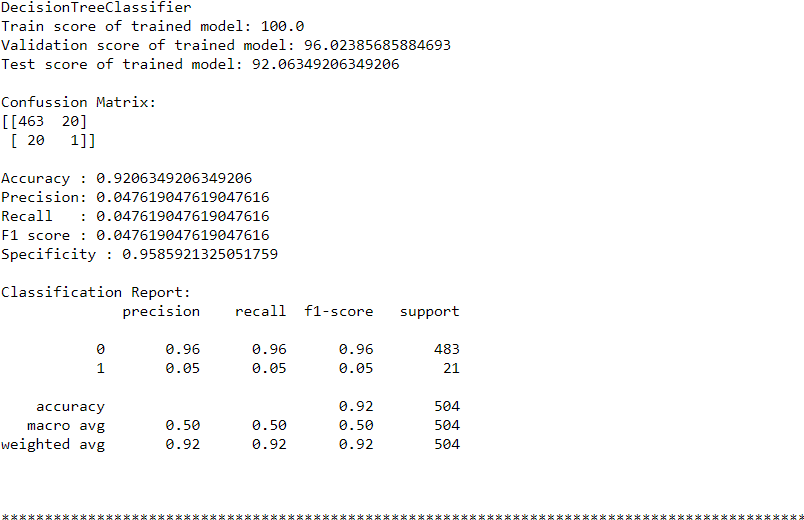


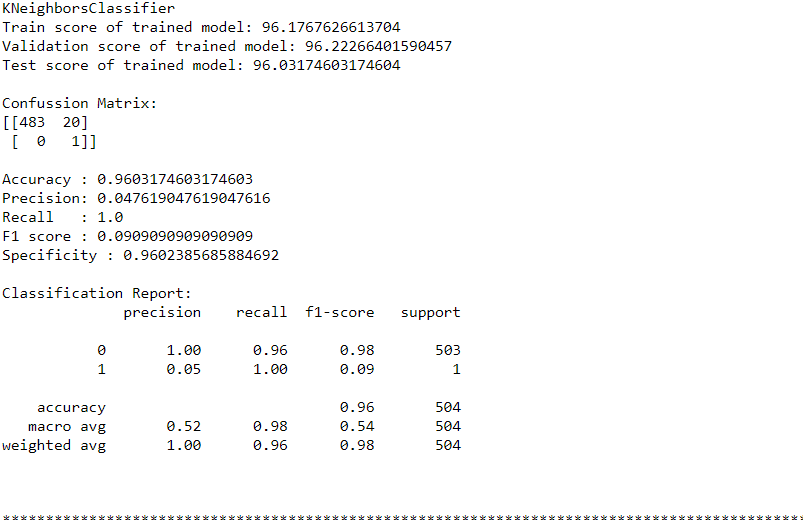


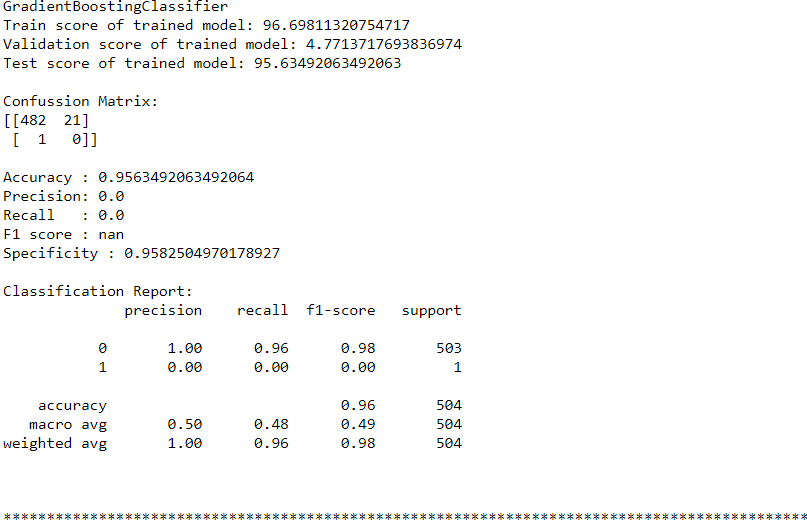


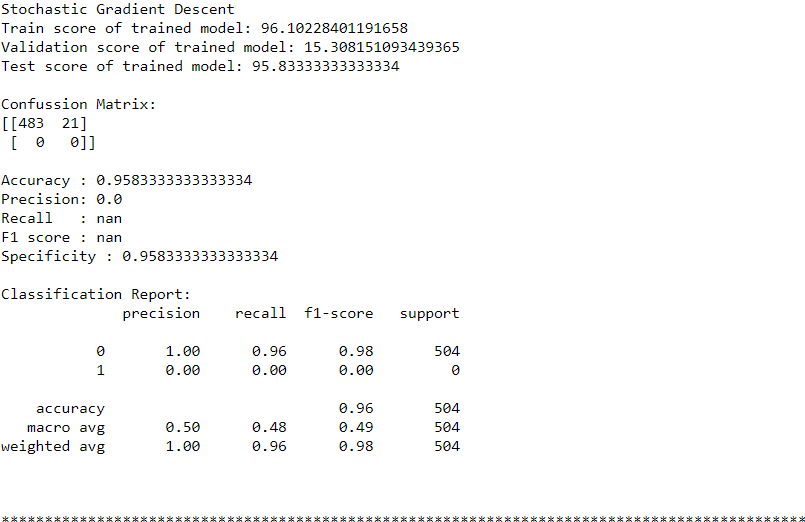












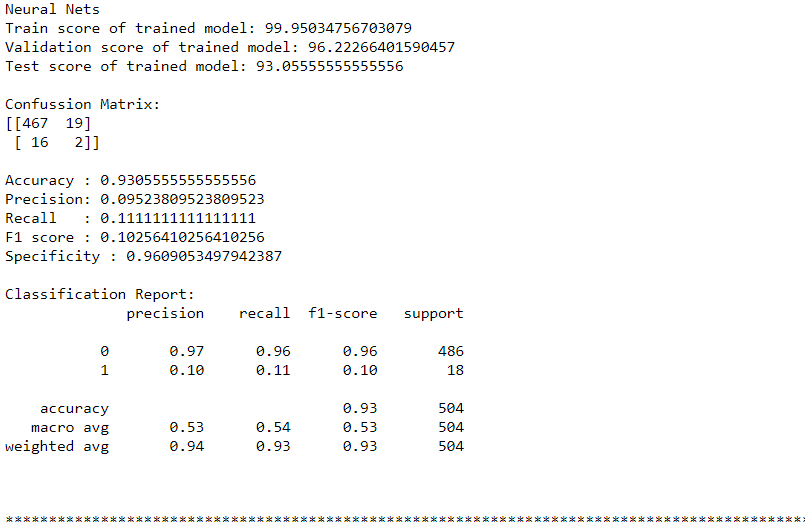


Figure 5.30: Output containing accuracy, precision, recall, F1 score, specificity and confusion matrix of various ML algorithms



Figure 5.31: Code snippet to graphically visualize the train-validation-test score of models



Figure 5.32: Graphical output of train-validation- test score of models

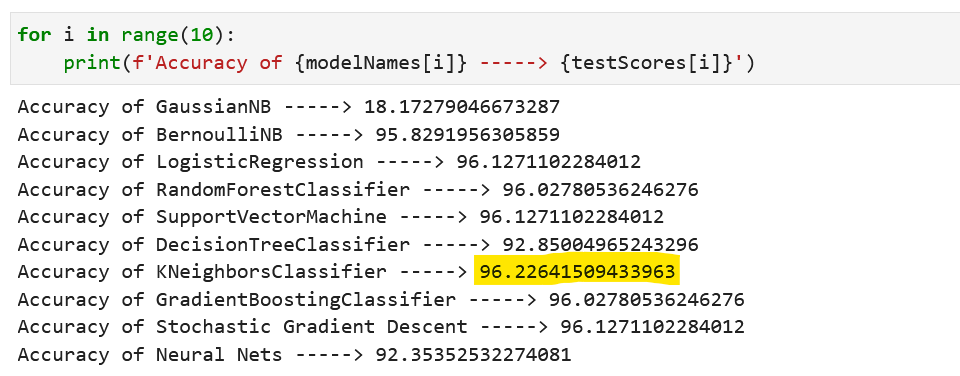


Figure 5.33: Code snippet and output showing the accuracy of each model

We observed that K Neighbors Classifier has the highest accuracy of 96.23%. Figure 5.14 represents the detailed report of performance of each of the ML classifier we used on our dataset. Figure 5.15 represents a histogram representation of score of testing, validation and training obtained for each of the classifier and figure 5.16 represents the accuracy score obtained by each of the model. Figure 5.17 contains a tabular representation of scores of accuracies, precision, recall and F1 score obtained by each model. Based on these results we have selected KNN as the classifier giving best performance on our dataset.

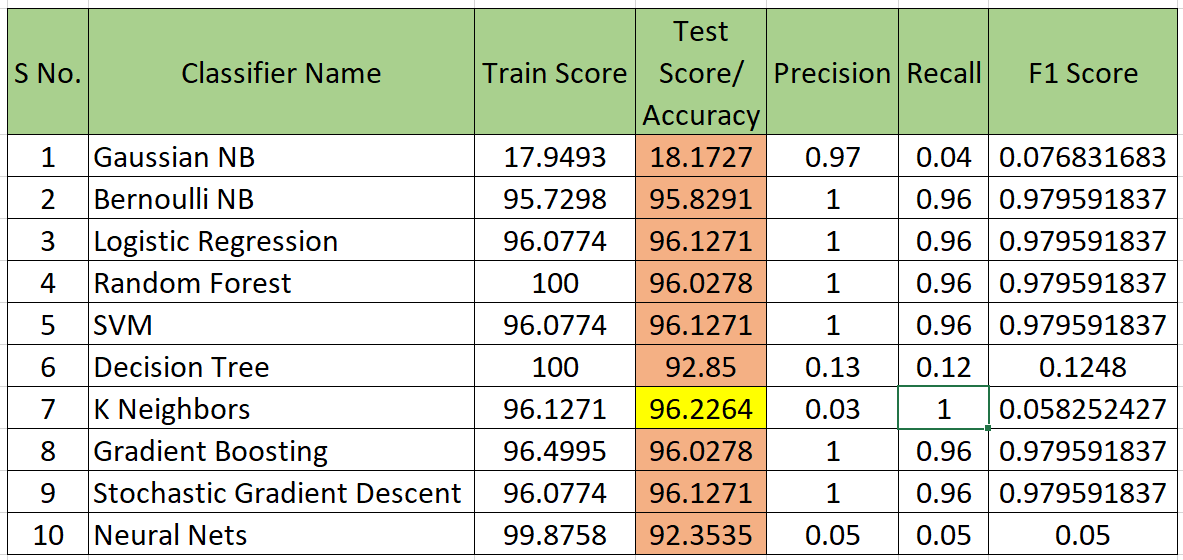


Figure 5.34: Comparison Score of different models representing scores of accuracies, precision, recall, and F1 score.

**5.11 MODEL SELECTION AND OPTIMIZATION**

We evaluated these models according to their accuracies, precision, recall, F1 score, specificity with the help of their confusion matrix. Selecting model based on parameter accuracy, Best algorithm is KNN with highest accuracy of 96.03%. Now we will make k-Fold Cross Validation and Hyper-Parameter Optimization for KNN algorithm**.**

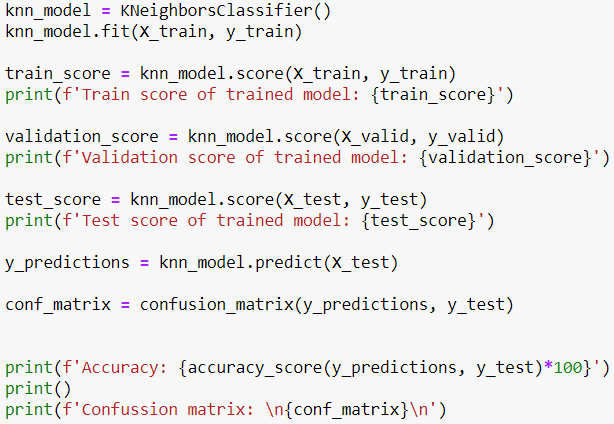
****

Figure 5.35: Code snippet for model selection and training for optimization

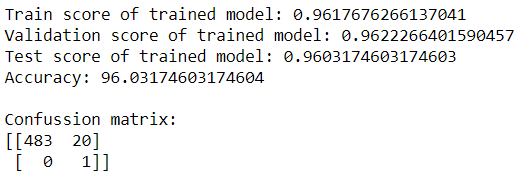
****

Figure 5.36: K Neighbors Classifier score along with accuracy and confusion matrix

**5.11.1 SAVING THE BEST MODEL**

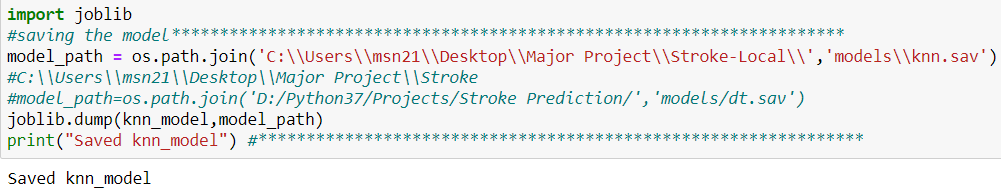


Figure 5.37: Code snippet for saving the KNN Model

**5.11.2 MODEL OPTIMIZATION: K-FOLD CROSS VALIDATION**

Cross Validation will enable us to see whether we are facing an overfitting problem and also to see the quality of our model. Thus, it will enable us to test the performance of our model before encountering high error rates in the test data set that we have not seen yet. It is a method that is frequently used because it is easy to apply. Figure 5.17 represent how changing the value of “k” in k-fold validation is affecting the cross-validation score. It shows that cross validation score has least value for k = 4 and achieved highest value on multiple values of k. It shows that our model is not overfit on any particular set or section of data.

**cv = 10 means k = 10 for KNN.**

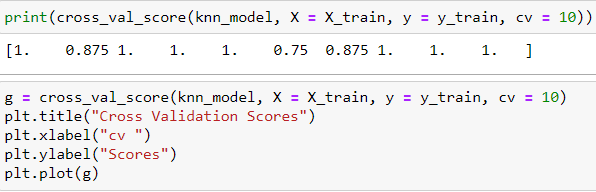


Figure 5.38: Code snippet to implement K-Fold Cross validation

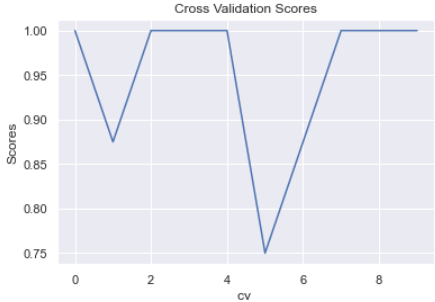


Figure 5.39: Graphical output after implementing K-Fold Cross validation

**5.11.3 MODEL OPTIMIZATION: HYPERPARAMETER OPTIMIZATION**

Unlike parameters, hyperparameters are not learned during training the model. They are determined by the data scientist before the modeling phase. For example, KNN algorithm, which is one of the non-parametric classification algorithms, makes classification by looking at the nearest k neighbors to the desired value. Here, the k number (n\_neighbors:) and the distance metric (metric:) to be used are the hyperparameters that should be specified by the data scientist before the modeling, which increases the performance of the model.

**Hyperparameter optimization** is the process of finding the most suitable hyperparameter combination according to the success metric specified for a machine learning algorithm.

Given that there are dozens of hyperparameters for a machine learning algorithm and dozens of values these hyperparameters can take, it's clear how difficult it will be to try all combinations one by one and pick the best combination. For this reason, different methods have been developed for hyperparameter optimization. GridSearchCV and RandomizedSearchCV are among these methods.

### **GridSearchCV:** For the hyperparameters and their values that are desired to be tested in the model, a separate model is established with all combinations and the most successful hyperparameter set is determined according to the specified metric. [43][45]

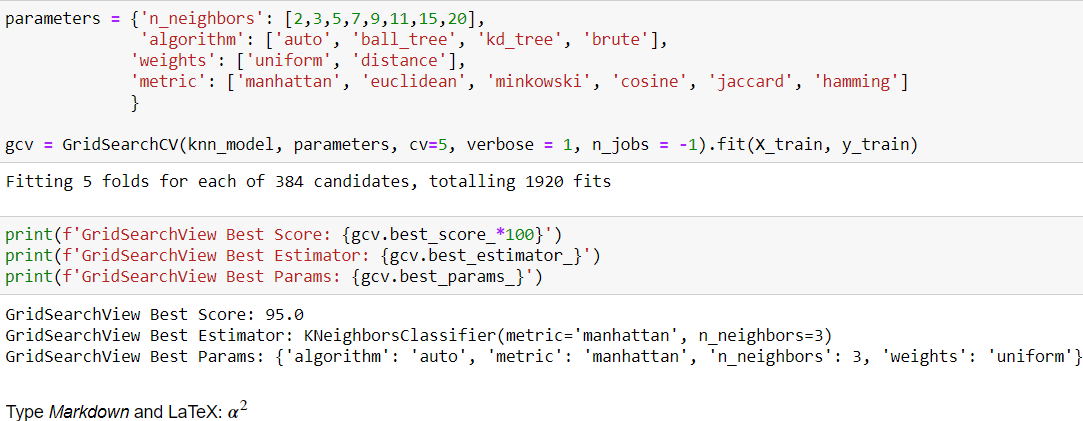


Figure 5.40: Code snippet and sample output after applying GridSearchCV

### **RandomizedSearchCV:** A set of hyperparameters is randomly selected and tested by cross-validation and the model set up. These steps continue until the specified calculation time limit or the number of iterations is reached. [44]



Figure 5.41: Code snippet and sample output after applying RandomizedSearchCV

**5.11.4 BEST FEATURE SELECTION**



Figure 5.42: Code snippet for best feature selections

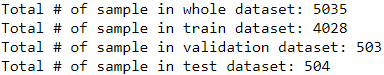


Figure 5.43: Sample output showing train, validation and test dataset

**5.12 WEB PAGE DEVELOPMENT**

For the purpose of providing interfaces to users, we needed to develop a platform. Thus, we chose web development for this. In this project, we created a web app using Python and its web framework flask, which reduces development time and allows us to build faster and smarter. For the Frontend, we have created a simple webpage which consist of a form which is to be filled by the user. Figure 5.18 represents the homepage with different attributes and their constraints mentioned along with them.

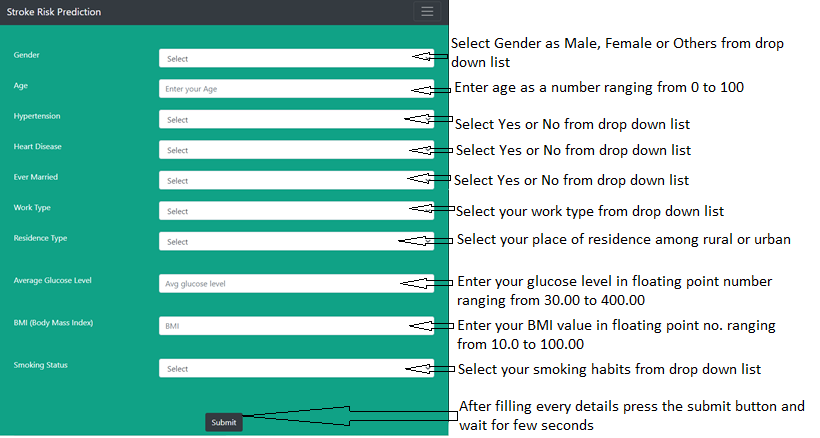


Figure 5.44: Layout of the webpage (User form)

In figure 5.18 user needs to fill all the fields as all the fields are compulsory. Each of the field is provided with a constraint check to prevent user from entering any random value. The above picture also shows the range and datatype of inputs each field can take. Drop down list is provided with gender, hypertension, heart disease, work type, residence type, marital status and smoking habits to prevent random values and guide user to select one among the available options. These options are linked with a corresponding numerical value by the backend flask model and passed to the prediction model for predicting the likelihood of stroke.

**5.13 CONNECTING MODEL WITH FRONTEND AND PYTHON CODE**

The backend and frontend both work together to serve a single goal. It’s pretty helpful to keep it in mind at all times. They are made, so a user can access them.

WORKING OF WEBSITE

The user points their browser to the website’s URLs and waits for the browser to render the page. The user sees a useful and usable page. The user interacts with the page.

Thus, till now, our website was working as static. In order to connect it to our working python code, we used flask i.e., micro-framework of python. In this, we created several routes using route decorator and thus it helped us in hosting it to a local server - http://localhost:7000/

**5.14 PREDICTING THE RISK OF STROKE**

The index() function is decorated with @app.route so that it is invoked when the browser sends a POST request. Using the request module of Flask, it creates an object of the file and saves it in local storage. At the same time, it sends that filled parameters to python code by calling its function which in response returns the output string, which is then displayed on the webpage.

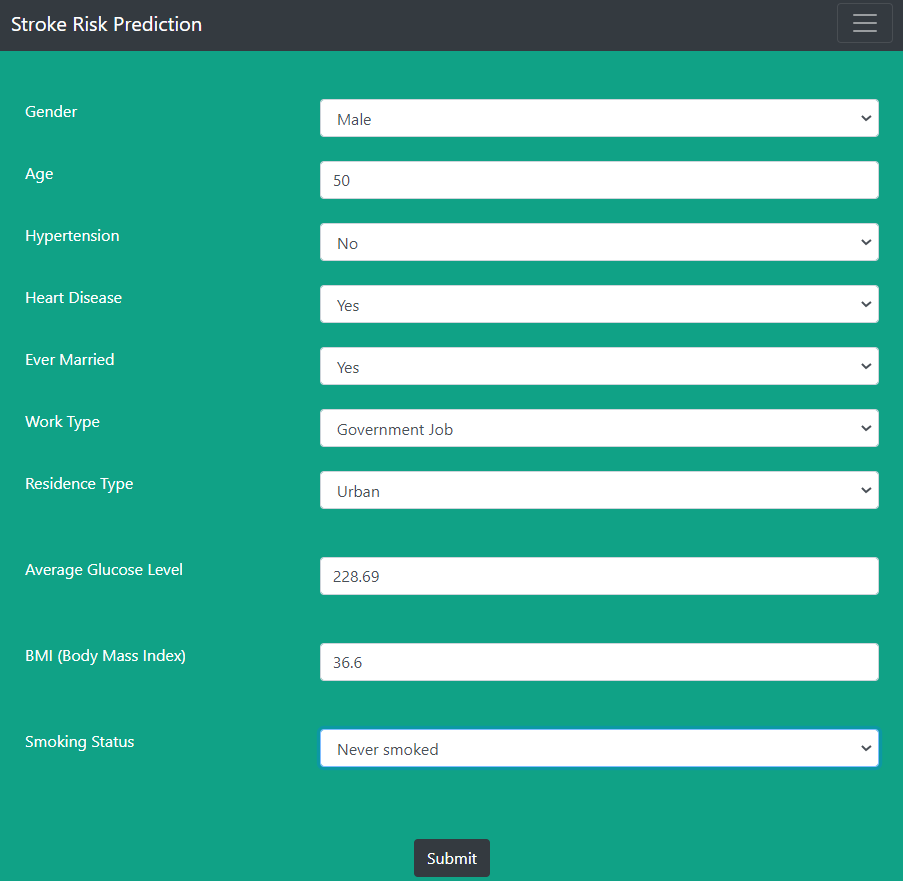


Figure 5.44a: Sample input data filled in the Form

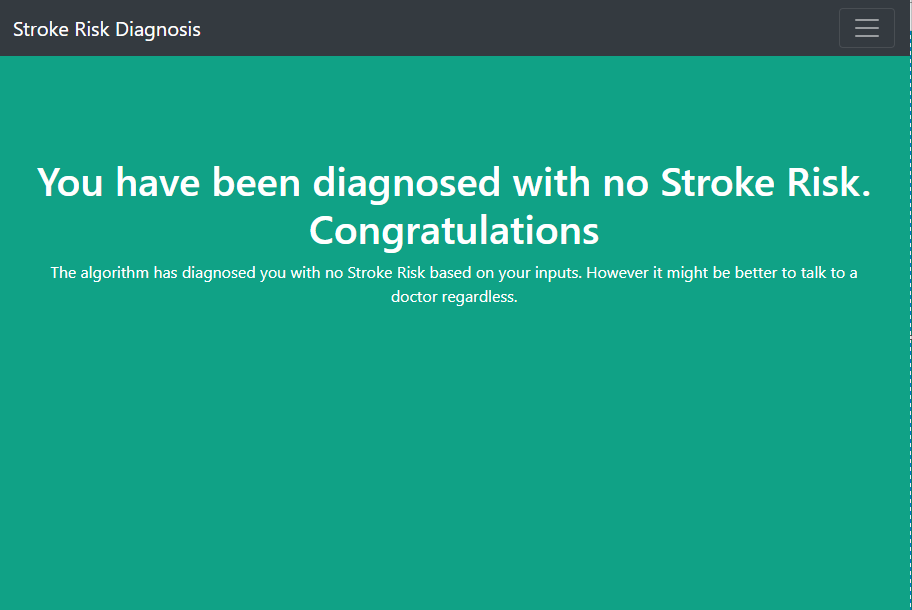


Figure 5.44b: Sample results displayed at the Web Platform when using the model for prediction

Figure 5.18 shows the front page of the Web App as visible to the user. Here he/she can fill the form based on their medical history and click on the submit button. A sample of filled detailed is provided in figure 5.19a. The model functions in the background and provide the stroke risk which is shown on the Web App as shown in figure 5.19b. Flask library is used to fetch the result predicted by the model and display it on the Web Page. If the model predicts as “You have diagnosed with no stroke risk” it shows that the user who has entered the details has rare likelihood of stroke. And if the result says “You have diagnosed with Stroke risk” it means that the user who has entered the details has high likelihood of stroke. In this scenario we recommend them to visit to a doctor for consultation. Otherwise, also we recommend users to undergo a periodic check for stroke risk if they have any of the symptoms. We believe this project will provide users with ease of testing the chances of stroke to them. And if further tested and developed with huge dataset, it holds real life application in clinical primary trails.

**CHAPTER – 6**

**CONCLUSIONS AND FUTURE SCOPE**

**6.1 CONCLUSIONS**

Through this project, we were able to build a system that can be used by user from the timing, location of his/her choice and check for chances of stroke to him. We compared 10 different machine learning classifiers based on accuracy on our dataset, used the best one and performed k-fold validation and hyper-parameter optimization on it too. We were able to understand the different and working of different classifiers and reason behind their particular behavior on our dataset. We are ultimately able to achieve our objective of learning about various machine learning classifier, comparing them and using the best one to build a system which can be used by anyone from anywhere to test the chances of stroke to him/her.

**6.2 FUTURE SCOPE**

There is a huge feature approach for this model some of which are enlisted below:

* The model is currently trained and tested for accuracy on a small dataset of approx. 5k records. We believe training it on any other dataset, which is bigger in size would enhance or prediction capability even more.
* Right now, the user is only asked for details in the form. We plan to enhance the web platform to display more information and statistics about the stroke.
* An android app for same purpose can be really useful and handy, we will this could be a potential future scope.
* Right now, the model uses 10 parameters for predicting the chances of stroke, adding more crucial parameters which play role in stroke and its prediction could be a really nice enhancement to the model.
* Adding more information about what stroke actually is, it causes, symptoms and other related and relevant information on the web page is also a nice and useful scope.
* Adding functionality to fetch all health record data and then use patient history for predicting chances of stroke could be a super huge enhancement.

**CHAPTER - 7**

**REFERENCES**

**Datasets:**

[1] “Dataset: Cardiovascular health study (CHS).” https://biolincc.nhlbi.nih.gov/studies/chs/. Accessed: 2016-05-8.

[2] "Stroke Prediction Dataset - (fedesoriano)." https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

**Model Classifiers: -**

[1] Stroke stats from: - <https://www.cdc.gov/stroke/facts.htm>

[2] Wikipedia Stroke page: - https://en.wikipedia.org/wiki/Stroke

[3] Scikit-learn Supervised Models - https://scikit-learn.org/stable/supervised\_learning.html

[4] Tony F. Chan, Gene H. Golub, Randall J. LeVeque: -Dept. Of CS, Stanford NOV,1979 – “GaussianNB - UPDATING FORMULAE AND A PAIR-WISE ALGORITHM FOR COMPUTING SAMPLE VARIANCES” - {<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html> }

[5] MultinomialNB - {<https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html> } {<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html> }

[6] BernoulliNB - {<https://nlp.stanford.edu/IR-book/html/htmledition/the-bernoulli-model-1.html> } {<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html> }

[7] CategoricalNB - {<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html> }

[8] LinearRegression - {<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html> }

[9] KNeighborsClassifier - {<https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm> } {<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html> }

[10] Mark Schmidt, Nicolas Le Roux, Francis Bach - 10 May 2016 – “Logistic Regression - Minimizing Finite Sums with the Stochastic Average Gradient” - {<https://hal.inria.fr/hal-00860051v2> }

[11] Logistic Regression - {<https://en.wikipedia.org/wiki/Logistic_regression> } {<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html> }

[12] Aaron Defazio, Francis Bach, Simon Lacoste-Julien - 16 Dec 2014 - “Logistic Regression - SAGA: A Fast Incremental Gradient Method with Support for Non-Strongly Convex Composite Objectives” - {arXiv:1407.0202v3, {<https://arxiv.org/pdf/1407.0202.pdf> }

[13] Hsiang-Fu Yu, Fang-Lan Huang, Chih – Jen Lin – “Logistic Regression - Dual Coordinate Descent Methods for Logistic Regression and Maximum Entropy Models” - {<https://www.csie.ntu.edu.tw/~cjlin/papers/maxent_dual.pdf> }

[14] Francesco Mola- 1998 – “Decision Tree Classifier - Classification and Regression Trees Software and New Developments”- {<https://sci-hub.se/10.1007/978-3-642-72253-0_42> }

[15] Decision Tree Classifier - {<https://en.wikipedia.org/wiki/Decision_tree_learning> } {<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html> }

[16] Decision Tree Classifier - The Elements of statistical Learning eBook- {<https://web.stanford.edu/~hastie/Papers/ESLII.pdf> }

[17] Tarun Gupta - Oct 2020 – “Decision Tree Classifier - Understanding Decision Tree Classifier Blog” - {<https://towardsdatascience.com/understanding-decision-tree-classifier-7366224e033b> }

[18] Support Vector Classification (SVC) - {<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html> }

[19] Chih - Chung Chang and Chih Jen Lin - Jan 2021 – “SVC - LIBSVM: A Library for Support Vector Machines”- {<https://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf> }

[20] John C. Platt - 1999 - “SVM - Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”- {<http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.1639> }

[21] Avinash Navlani - Dec 2019 - “SVM - Support Vector Machines with Scikit-learn Blog” {<https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python>

[22] KMeans- {<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html> }

[23] KMeans Clustering - {<https://en.wikipedia.org/wiki/K-means_clustering> }

[24] Imad Dabbura - Sept 2018 – “KMeans - K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks” - {<https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a> }

[25] Random Forest Classifier - {<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> }

[26] LEO BREIMAN - 2001 – “Random Forest Classifier - Random Forest” - {<https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf> }

[27] Random Forest Classifier - {<https://en.wikipedia.org/wiki/Random_forest> }

[28] Tony Yiu- Jun 2019 – “Random Forest Classifier - Understanding Random Forest - How the Algorithm Works and Why It Is So Effective Blog” - {<https://towardsdatascience.com/understanding-random-forest-58381e0602d2> }

[29] Avinash Navlani - May 2018 - “Random Forest Classifier - Understanding Random Forests Classifiers in Python Blog” - {<https://www.datacamp.com/community/tutorials/random-forests-classifier-python> }

[30] Gradient Boosting Classifier - {<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html> }

[31] StandardScaler - {<https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html> }

[32] Jason Brownlee - Aug 2016- “XGBClassifier - How to Develop Your First XGBoost Model in Python Blog” - {<https://machinelearningmastery.com/develop-first-xgboost-model-python-scikit-learn/> }

[33] Jason Brownlee - May 2018 – “k-Fold Cross Validation - A Gentle Introduction to k-fold Cross-Validation Blog” {<https://machinelearningmastery.com/k-fold-cross-validation/> }

[34] k-Fold Cross Validation - Cross-validation: evaluating estimator performance - {<https://scikit-learn.org/stable/modules/cross_validation.html> }

[35] Markus Ojala, Gemma C. Garriga - Jun 2010 – “k-Fold Cross Validation - Permutation Tests for Studying Classifier Performance” - {<https://www.jmlr.org/papers/volume11/ojala10a/ojala10a.pdf> }

[36] k-Fold Cross Validation - {<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html> }

[37] Sanjay M- Nov 2018 – “k-Fold Cross Validation - Why and how to Cross Validate a Model? Importance and types of Cross-validation techniques Blog”- {<https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f> }

[38] Hyper-Parameter Optimization - {<https://en.wikipedia.org/wiki/Hyperparameter_optimization> }

[39] Pier Paolo Ippolito - Sept,2019 – “Hyper-Parameter Optimization - An introduction on how to fine-tune Machine and Deep Learning models using techniques such as: Random Search, Automated Hyperparameter Tuning and Artificial Neural Networks Tuning” - {<https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d> }

[40] James Bergstra, Yoshua Bengio - 2012 – “Hyper-Parameter Optimization - Random Search for Hyper-Parameter Optimization” - {<https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a> }

[41] James Bergstra, Yoshua Bengio, Remi Bardenet, Balazs Kegl - Nov,2011 – “Hyper-Parameter Optimization - Algorithms for Hyper-Parameter Optimization” - {<https://hal.inria.fr/hal-00642998/file/draft1.pdf> }

[42] Yufei Xiaa, Chuanzhe Liua, Yu Ying Lib, Nana Liua - Feb,2017 – “Hyper-Parameter Optimization - A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring” - {<https://sci-hub.se/10.1016/j.eswa.2017.02.017> }

[43] GridSearchCV - {<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html> }

[44] RandomizedSearchCV - {<https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html> }

[45] Papa Moryba Kouate - Sept,2020 – “GridSearchCV - Machine Learning: GridSearchCV & RandomizedSearchCV - Hyperparameter tuning with GridSearchCV and RandomizedSearchCV Blog” - {<https://towardsdatascience.com/machine-learning-gridsearchcv-randomizedsearchcv-d36b89231b10>}

[46] Benjamin Letham, Cynthia Rudin, Tyler H. Mccormick And David Madigan – 2015- “Interpretable Classifiers Using Rules And Bayesian Analysis: Building A Better Stroke Prediction Model”

[47] Luis Garcia-Terriza, Jose L. Risco-Martin, Jose L. Ayala, Gemma Reig Rosello, Juan M. Camarasaltas - April 2019 – “Comparison of Different Machine Learning Approaches to Model Stroke Subtype Classification and Risk Prediction”

[48] Ibrahim Abuaqel, Abdullah Alsaif, Mohammed Almulhim, Mohammed Aljallal, Musab Alghuraibi, Mujahid Alzahrani, Abdulrahman AlHarby, Sunday O. Olatunji. - 2017

“Prediction Models Aided Postoperative Decision Making Based on Neural Network and Support Vector Machines”

[49] Pattanapong Chantamit-o-pas and Madhu Goyal – 2017 – “Prediction of Stroke Using Deep Learning Model”

[50] Priya Govindarajan, Ravichandran Kattur Soundarapandian, Amir H. Gandomi, Rizwan Patan, Premaladha Jayaraman, Ramachandran Manikandan - Nov 2018- “Classification of stroke disease using machine learning algorithms”

[51] Aditya Khosla, Yu Cao, Cliff Chiung-Yu Lin, Hsu-Kuang Chiu, Junling Hu, Honglak Lee - July 2017 – “An Integrated Machine Learning Approach to Stroke Prediction”

[52] Yu, J., Park, S., Lee, H., Pyo, C.S., Lee, Y.S. (2020). An elderly health monitoring system using machine learning and in-depth analysis techniques on the NIH stroke scale. Mathematics, 8(7): 1-16. <https://doi.org/10.3390/math8071115>

[53] Monteiro, M., Fonseca, A.C., Freitas, A.T., Melo, T.P., Francisco, A.P., Ferro, J.M., Oliveira, A.L. (2018). Using machine learning to improve the prediction of functional outcome in ischemic stroke patients. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 15(6): 1953-1959. <https://doi.org/10.1109/TCBB.2018.2811471>

[54] Sung, S.F., Lin, C.Y., Hu, Y.H. (2020). EMR-based phenotyping of ischemic stroke using supervised machine learning and text mining techniques. IEEE Journal of Biomedical and Health Informatics, 24(10):2922-2931. <https://doi.org/10.1109/JBHI.2020.2976931>

[55] Xie, Y., Jiang, B., Gong, E., Li, Y., Zhu, G., Michel, P., Wintermark, M., Zaharchuk, Z. (2019). Use of gradient boosting machine learning to predict patient outcome in acute ischemic stroke on the basis of imaging, demographic, and clinical information. American Journal of Roentgenology, 212(1): 44-51. <https://doi.org/10.2214/AJR.18.20260>

[56] Wang, F., Huang, Y., Xia, Y., Zhang, W., Fang, K., Zhou, X., Yu, X., Cheng, X., Li, G., Wang, X., Luo, G., Wu, D., Liu, X., Campbell, B.C.V., Dong, Q., Zhao, Y. (2020). Personalized risk prediction of symptomatic intracerebral hemorrhage after stroke thrombolysis using a machine-learning model. Therapeutic Advances in Neurological Disorder, 13: 1-10. <https://doi.org/10.1177/1756286420902358>

[57] Lin, C.H., Hsu, K.C., Johnson, K.R., Fann, Y.C., Tsai,

C.H., Sun, Y., Lien, L.M., Chang, W.L., Lin, C.L., Hsu, C.Y., Registry, T.S. (2020). Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry. Computer Methods and Programs in Biomedicine, 190: 105381. <https://doi.org/10.1016/j.cmpb.2020.105381>

[58] Sung, S.M., Kang, Y.J., Cho, H.J., Kim, N.R., Lee, S.M., Choi, B.K., Cho, G. (2020). Prediction of early neurological deterioration in acute minor ischemic stroke by machine learning algorithms. Clinical Neurology and Neurosurgery, 195: 105892. <https://doi.org/10.1016/j.clineuro.2020.105892>

[59] Bandi, V., Midhunchakkaravarthy, D., Bhattacharyya, D. (2020). Stroke Analysis. Mendeley Data. <https://doi.org/10.17632/jpb5tds9f6.1>

[60] P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi, R. Patan, P. Jayaraman, and R. Manikandan, “Classification of stroke disease using machine learning algorithms,” Neural Computing and Applications, vol. 32, no. 3, pp. 817–828, Feb. 2020.

[61] S. M. Reza, M. M. Rahman, and S. Al Mamun, “A new approach for road networks-a vehicle xml device collaboration with big data,” in 2014 International Conference on Electrical Engineering and Information & Communication Technology. IEEE, 2014, pp. 1–5.

[62] C.-A. Cheng, Y.-C. Lin, and H.-W. Chiu, “Prediction of the prognosis of ischemic stroke patients after intravenous thrombolysis using artificial neural networks,” Studies in Health Technology and Informatics, vol. 202, pp. 115–118, 2014.

[63] S. Cheon, J. Kim, and J. Lim, “The Use of Deep Learning to Predict Stroke Patient Mortality,” International Journal of Environmental Research and Public Health, vol. 16, no. 11, 2019.

[64] M. S. Singh and P. Choudhary, “Stroke prediction using artificial intelligence,” in 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), Aug. 2017, pp. 158–161.

[65] C. Chin, B. Lin, G. Wu, T. Weng, C. Yang, R. Su, and Y. Pan, “An automated early ischemic stroke detection system using CNN deep learning algorithm,” in 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST), Nov. 2017, iSSN: 2325-5994.

[66] M. Monteiro, A. C. Fonseca, A. T. Freitas, T. Pinho e Melo, A. P. Francisco, J. M. Ferro, and A. L. Oliveira, “Using Machine Learning to Improve the Prediction of Functional Outcome in Ischemic Stroke Patients,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15, pp. 1953–1959, Nov. 2018.

[67] T. Kansadub, S. Thammaboosadee, S. Kiattisin, and C. Jalayondeja, “Stroke risk prediction model based on demographic data,” in 2015 8th Biomedical Engineering International Conference (BMEiCON), Nov. 2015, pp. 1–3.

[68] S. Y. Adam, A. Yousif, and M. B. Bashir, “Classification of Ischemic Stroke using Machine Learning Algorithms,” International Journal of Computer Applications, vol. 149, no. 10, pp. 26–31, Sep. 2016.

[69] S.-F. Sung, C.-Y. Hsieh, Y.-H. Kao Yang, H.-J. Lin, C.-H. Chen, Y.- W. Chen, and Y.-H. Hu, “Developing a stroke severity index based on administrative data was feasible using data mining techniques,” Journal of Clinical Epidemiology, vol. 68, no. 11, pp. 1292–1300, Nov. 2015.

[70] Top causes of death: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>

[71] Metrics: <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>

[72] Evaluation Metrics :- https://en.wikipedia.org/wiki/Precision\_and\_recall

[73] Evaluation Metrics F1 Score: <https://en.wikipedia.org/wiki/F-score>

[74] Evaluation Metrics Sensitivity and Specificity: <https://en.wikipedia.org/wiki/Sensitivity_and_specificity#:~:text=Specificity%20(True%20Negative%20rate)%20measures,as%20not%20having%20the%20condition).>

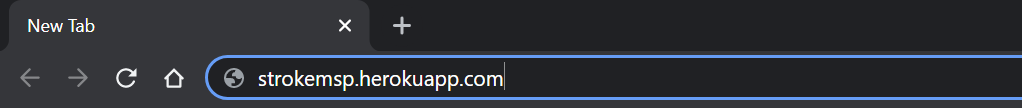
**APPENDICES**

**APPENDIX 1**

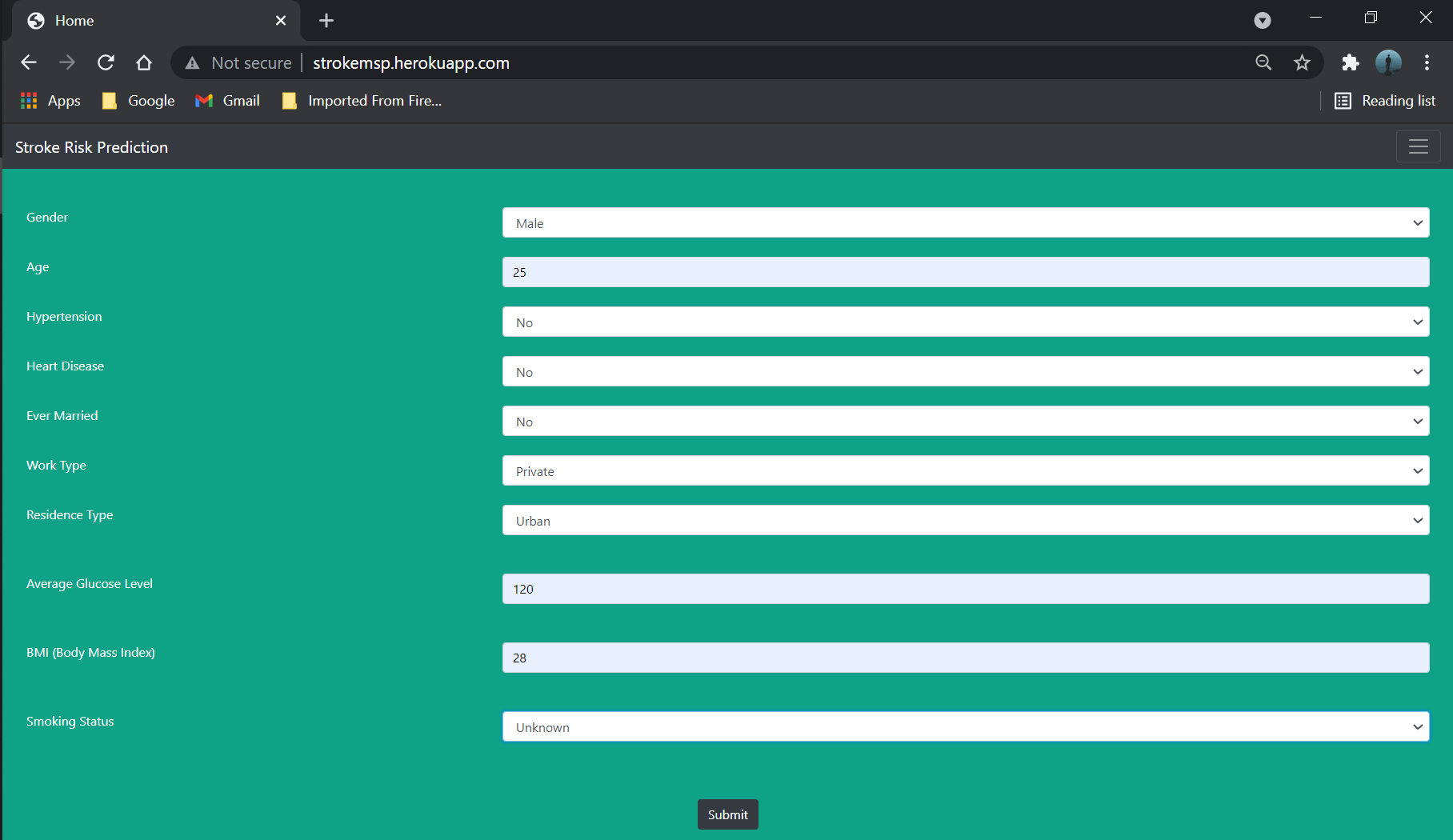
**1.1 USER GUIDE**

The user needs to follow the below-mentioned procedure for using the model to predict the chances of stroke to him/her.

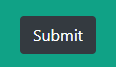
Step 1: Visit to the webpage using <https://strokemsp.herokuapp.com>



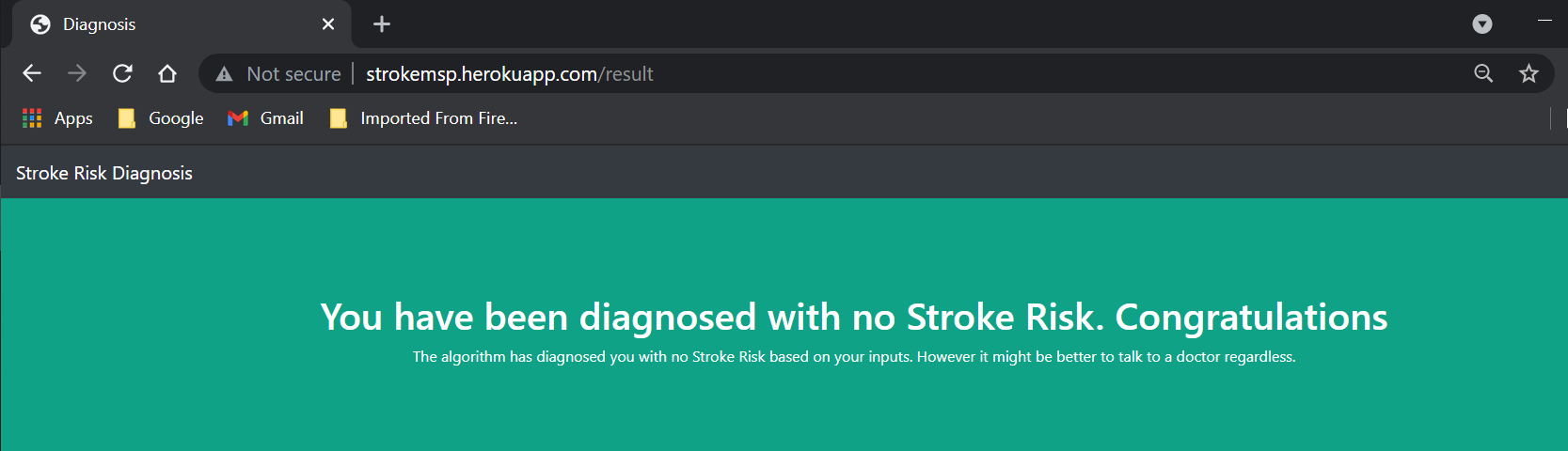
Step 2: Fill in all the details asked.



Step 3: Click on submit button and wait for a moment for the model to predict.



Step 4: Within a few moments the model will predict the results which is being displayed to you on a new web page.



Step 5: If the model predicts you to have stroke risk, we recommend you to pay a visit to a doctor.

Step 6: Done. If the model predicts you having no risk of stroke; and still, you feel any of the symptoms we recommend you to talk to a doctor.

**\*\*EOF\*\***