A MINI PROJECT REPORT ON

ANALYZING PERFORMANCE OF STROKE PREDICTION USING MACHINE LEARNING ALGORITHMS

SUBMITTED IN PARTIAL FULFILLMENT FOR THE

DEGREE OF BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND TECHNOLOGY

by

09 - Twinkle Bothara

10 - Rashi Chaturvedi

14 - Shruti Chavan

UNDER THE GUIDANCE OF

Prof. Monica Charate

USHA MITTAL INSTITUTE OF TECHNOLOGY

S.N.D.T. WOMEN'S UNIVERSITY

MUMBAI – 400049

2021 – 2022.

CERTIFICATE

This is to certify that Twinkle Bothara, Rashi Chaturvedi, Shruti Chavan has successfully completed seminar work on Analyzing Performance of Stroke Prediction Using Machine Learning Algorithms in the partial fulfillment for the bachelor's degree in Computer Science and Technology during the year 2021-2022 as prescribed by SNDT Women's University.

GUIDE

Prof. Monica Charate

HEAD OF THE DEPARTMENT

Prof. Kumud Wasnik

PRINCIPAL

Dr. Shikha Nema

Examinar 1

Examinar 2

ACKNOWLEDGEMENT

It is indeed a great pleasure and proud opportunity for us to present this mini project report for third year degree at Usha Mittal Institute of Technology. The success of this project has throughout depended upon a combination of hard work and an unending co-operation and guidance provided to us by our project guide. Ultimately no words could describe the deep sense of co-operation and ready nature to help us.

We would like to thank, **Ms.SHIKHA NEMA (Principal), Prof. KUMUD WASNIK (H.O.D. of CST Department), Prof.MONICA CHARATE (Project Guide)**, who made very valuable guidance and co-operation during our project.

Further we are thankful to all the teaching and non-teaching staff of Computer Science and Technology Department for their co-operation during the project work. We are very grateful to those who in the form of books had conveyed guidance in this project work.

Contents

1 INTRODUCTION **7**

2 PROBLEM STATEMENT **9**

3 LITERATURE SURVEY **11**

4 EXISTING SYSTEM **15**

5 PROPOSED SYSTEM **17**

6 SYSTEM ANALYSIS **18**

6.1 Dataset: 18

6.2 Preprocessing: 18

6.3 Proposed Algorithm: 19

7 SYSTEM REQUIREMENT **20**

7.1 Machine Learning Methodology 20

7.1.1 Decision Tree 21

7.1.2 Logistic Regression 21

7.1.3 Random Forest 22

7.1.4 Support Vector Machine 22

7.1.5 Gaussian Naive Bayes 24

7.1.6 LightGBM (Light Gradient Boosting Machine) 25

7.1.7 XGBoost 27

7.1.8 K-Nearest Neighbor 28

7.1.9 Stochastic Gradient Descent (SGD) 29

7.2 ASSET VISUALIZATION 30

7.2.1 Non-Functional Requirement 30

8 SYSTEM DEVELOPMENT METHODOLOGY 33

9 MODEL PHASES 34

10 DESIGN USING UML 36

11 TOOLS 39

12 SOFTWARE REQUIREMENT 39

13 HARDWARE REQUIREMENT 39

14 Implementation 41

15 FUTURE SCOPE 43

List of Figures

1 Existing system 15

2 State diagram 17

3 waterfall 34

4 Architectural 36

5 Flow chart 37

6 Work flow 38

7 Dataset loaded 41

8 Data info 42

1 INTRODUCTION

Stroke is a condition where the blood supply to the brain is disrupted, resulting in oxygen starvation, brain damage and loss of function. It is most frequently caused by a clot in an artery supplying blood to the brain, a situation known as ischemia. It can also be caused by hemorrhage when a burst vessel causes blood to leak into the brain. Stroke can cause permanent damage, including partial paralysis and impairment in speech, comprehension and memory. The extent and location of the damage determines the severity of the stroke, which can range from minimal to catastrophic. According to the World Health Organization, every year 15 million people worldwide suffer a stroke.Stroke is uncommon in people under 40 years; when it does occur, the main cause is high blood pressure. However, stroke also occurs in about 8% of children with sickle cell disease.

Diagnosis is a complicated and important task that needs to be executed accurately and efficiently. The diagnosis is often made, based on the doctor's experience & knowledge. This leads to unwanted results & excessive medical costs of treatments provided to patients. The early diagnosis of stroke plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduces the complications. This project aims to predict future strokes by analyzing data of patients which classifies whether they will have a stroke or not using machine-learning algorithms. With the advancement of technology in the medical field, predicting the occurrence of a stroke can be made using Machine Learning. The algorithms present in Machine Learning are constructive in making an accurate prediction and give correct analysis.

The major motivation behind this research-based project was to explore the

feature selection methods, data preparation and processing behind the training models in machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Hence this project is carried out with the motivation to explore behind the models, and further implement various models to train the obtained data. Furthermore, as the whole machine learning is motivated to develop an appropriate computer-based system and decision support that can aid early detection of heart disease.

2 PROBLEM STATEMENT

Stroke is the second leading cause of death worldwide and remains an important health burden both for the individuals and for the national healthcare systems. Potentially modifiable risk factors for stroke include hypertension, cardiac disease, diabetes, and dis-regulation of glucose metabolism, atrial fibrillation, and lifestyle factors.

The major challenge in strokes is its detection. There are instruments available which can predict strokes but either they are expensive or are not efficient to calculate chances of strokes in humans. Early detection of strokes can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more patience, time and expertise. But with the right treatment, the symptoms of strokes can be reduced and the functioning of the heart improved. The predicted results can be used to prevent and thus reduce cost for surgical treatment and other expenses.

Decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden in the data set and databases. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients.

The overall objective of my work will be to predict accurately with few tests and attribute the presence of heart disease. Attributes considered form the primary basis for tests and give accurate results more or less. Many more input attributes can be taken but our goal is to predict with few attributes

and faster efficiency the risk of having heart disease. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data. Then we intend to develop the application to provide a personalized warning on the basis of each user's level of stroke risk and a lifestyle correction message about the stroke risk factors.

3 LITERATURE SURVEY

Many researchers have already used machine learning based approaches to predict strokes.This section completely discusses the related work done by different researchers in the same area of research.

Govindarajan et al. conducted a study to categorize stroke disorder using a text mining combination and a machine learning classifier and collected data for 507 patients. For their analysis, they used various machine learning approaches for training purposes using ANN, and the SGD algorithm gave them the best value, which was 95%

Yu et al. 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 analyzing 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. The conclusion is that in this work, the C4.5 decision tree algorithm has made promising results in determining the criticality and analyzing the classification of the stroke. Also it had decreased the factors of the stroke from the database of NIHSS features. Based on the hypothetical solutions 91.11% decent accuracy is obtained through the C4.5 decision tree algorithm.

Monteiro et al. 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 cause of disorder and death all over the world among people of 65 years and in adults. The proposed methodology 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% accuracy 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. have proposed a methodology that can be examined for automated phenotyping by further classifying the ischemic stroke into 4 subdivisions. This model depends 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 classified solutions helps in increasing the performance by taking 66% subjects on training and 34%

subjects on testing.

Xie et al. have proposed a model to combine common stroke biomarkers by developing machine learning techniques and to analyze 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 rankin 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 five fold 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. have implemented a machine learning model in the configuration of the risk of symptomatic intracerebral hemorrhage (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 analyze 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. 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. have implemented machine learning algorithms to analyze the stroke outcome with acute minor stroke. Among 739 patients, 61 patients had 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 Rankin scale score of ≥ 2. In this work, four machine learning models such as bootstrap decision forest, boosted trees, Logistic Regression, and Deep neural network were used in analyzing the early signs of neurological deterioration and examined with a decent accuracy of 94.6%.

4 EXISTING SYSTEM

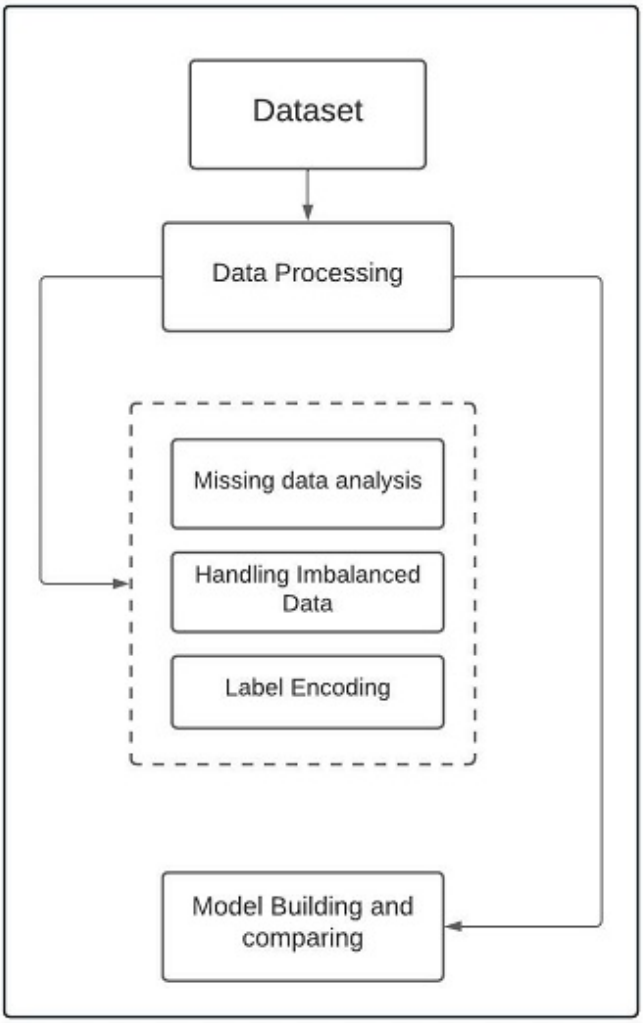


Figure 1: Existing system

With growing development in the field of medical science alongside machine learning various experiments and research has been carried out in recent years releasing the relevant significant papers. In these existing research works, the authors have done only a classification of various types of strokes. This leads to improper classification and we do not attain decent accuracy to predict the stroke severity.

Due to this, there is a gap identified for predicting the risk levels of stroke factors which are low, moderate, high and severe. To overcome this limitation, in this research work we have deployed three levels of risk identification hierarchy modules. These modules have been implemented with the help of the proposed algorithm to predict the risk levels of stroke factors along with classification.

5 PROPOSED SYSTEM

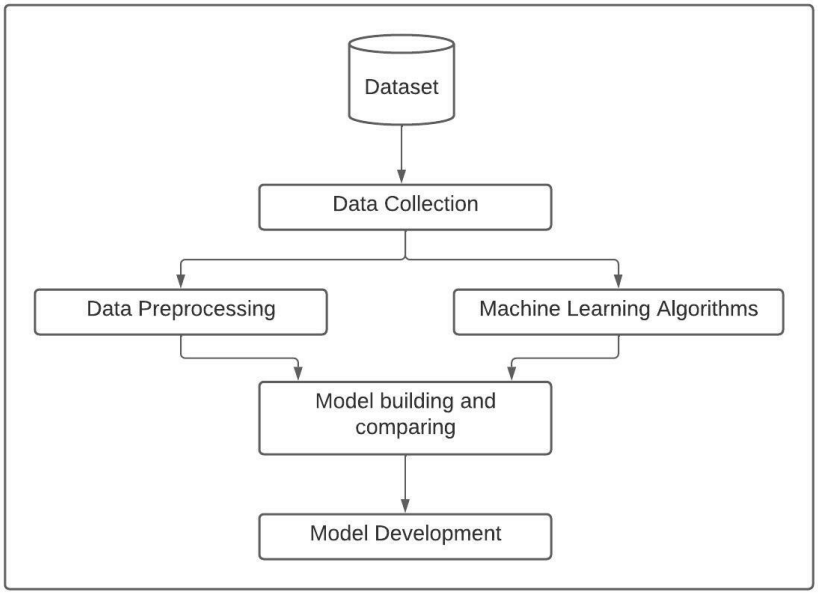


Figure 2: State diagram

The proposed work predicts stroke by exploring the five algorithms: SVM, Decision tree, Random Forest, K-nearest neighbor and Logistic regression and does performance analysis. The data has become available for model construction once it has been processed. A preprocessed dataset and machine learning techniques are needed for the model construction. The data has become available for model construction once it has been processed. A preprocessed dataset and machine learning techniques are needed for the model construction. After creating five alternative models, the accuracy measures, namely accuracy score, precision score, recall score, and F1 score are used to compare them. The objective of this study is to effectively predict if the patient suffers from any kind of stroke. The health professional enters the input values from the patient's health report. The data is fed into a model which predicts the probability of having a heart stroke.

6 SYSTEM ANALYSIS

6.1 Dataset:

The stroke prediction dataset was used to perform the study. There were 43400 rows and 12 columns in this dataset. The value of the output column stroke is either 1 or 0. The number 0 indicates that no stroke risk was identified, while the value 1 indicates that a stroke risk was detected. The probability of 0 in the output column (stroke) exceeds the possibility of 1 in the same column in this dataset. 638 rows alone in the stroke column have the value 1, whereas 29470 rows have the value 0. To improve accuracy, data preprocessing is used to balance the data.

6.2 Preprocessing:

Before building a model, data preprocessing is required to remove unwanted noise and outliers from the dataset that could lead the model to depart from its intended training. This stage addresses everything that prevents the model from functioning more efficiently. Following the collection of the relevant dataset, the data must be cleaned and prepared for model development. As stated before, the dataset used has twelve characteristics. To begin with, the column id is omitted since its presence has no bearing on model construction. The dataset is then inspected for null values and filled if any are detected. The null values in the column BMI are filled using the data column's mean in this case. Label encoding converts the dataset's string literals to integer values that the computer can comprehend. As the computer is frequently trained on numbers, the strings must be converted to integers. The gathered dataset has five columns of the data type string. All strings are encoded during label encoding, and the whole dataset is transformed

into a collection of numbers. The dataset used for stroke prediction is very imbalanced. The dataset has a total of 43400 rows, with 638 rows indicating the possibility of a stroke and 29470 rows confirming the lack of a stroke. While using such data to train a machine-level model may result in accuracy, other accuracy measures such as precision and recall are inadequate. If such unbalanced data is not dealt with properly, the findings will be inaccurate, and the forecast will be ineffective. As a result, to obtain an efficient model, this unbalanced data must be dealt with first. To improve the accuracy and efficiency of this job, the data is divided into training and testing data with a ratio of 80 percent training data and 20 percent testing data. After splitting, the model is trained using a variety of classification methods. SVM, Decision tree, Random Forest, K-nearest neighbor and Logistic regression.

6.3 Proposed Algorithm:

The most common disease identified in the medical field is stroke, which is on the rise year after year. Using the publicly accessible stroke prediction dataset, the study measured ten commonly used machine learning methods for predicting brain stroke recurrence.

7 SYSTEM REQUIREMENT

A software requirements specification is a description of a software system to be developed, describing all the functional and nonfunctional requirements which also include a set of use cases that describe interactions between the users. In this system following are the functional requirements: -

- Machine Learning Methodology
- Asset Visualization

7.1 Machine Learning Methodology

Using this methodology, the modeler can discover the "performance ceiling" for the data set before settling on a model. In many cases, a range of models will be Prediction of Stroke Using equivalent in terms of performance so the practitioner can weigh the benefits of different methodologies.

Few methodologies used in our projects are:

- Decision Tree
- Logistic Regression
- Random Forest
- SVM
- Gaussian Naive Bayes(GNB)
- LightGBM(LGBM)
- XGboost(XGB)
- K Nearest Neighbour

- AdaBoost Classifier
- Stochastic Gradient Descent Classifier(SGD)

7.1.1 Decision Tree

A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Decision trees are one of the important methods for handling high dimensional data. Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods.

7.1.2 Logistic Regression

Logistic regression is a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for a given set of features(or inputs), X.

It measures the relationship between the dependent variable (TenyearCHD) and the one or more independent variables (risk factors) by estimating probabilities using the underlying logistic function (sigmoid function). Sigmoid function is used as a cost function to limit the hypothesis of logistic regression between 0 and 1 (squashing) i.e. $0 \leq h\theta(x) \leq 1$.

In logistic regression cost function is defined as:

$$cost(h\theta(x), y) = -log(h\theta(x)) \quad \text{if} \quad y = 1$$

$$-log(1 - h\theta(x)) \quad \text{if} \quad y = 0$$

Logistic Regression relies highly on the proper presentation of data. So, to make the model more powerful, important features from the available data set are selected using Backward elimination and recursive elimination techniques.

7.1.3 Random Forest

RFs are composed of numerous independent decision trees that were trained individually on a random sample of data. These trees are created during training, and the decision trees' outputs are collected. A process termed voting is used to determine the final forecast made by this algorithm. Each DT in this method must vote for one of the two output classes (in this case, stroke or no stroke). The final prediction is determined by the RF method, which chooses the class with the most votes.

The flexibility of the random forest is one of its most alluring features. It may be utilized for relapse detection and grouping tasks, and the overall weighting given to information characteristics is readily apparent. Additionally, it is a beneficial approach since the default hyperparameters it employs often give unambiguous expectations. Understanding the hyperparameters is critical since there are relatively few of them, to begin with. Overfitting is a well known problem in machine learning, although it occurs seldom with the arbitrary random forest classifier. If there are suffix efficient trees in the forest, the classifier will not overfit the model.

7.1.4 Support Vector Machine

The main aim of this model is to develop the exact linear way or deterministic partition which separates n-proportional space into groups such that

providing easy access of combining the data which is newly formed into their respective modules for further references. This type of sorting the data by an exact linear way can also be referred to as hyperplane. Since considering these outermost vector points that are supportive in building the hyperplane are termed as support points and so the algorithm is named as Support vector algorithm. With the help of the demo graphs that are categorized into two variant ways which are divided considering the deterministic partition or a hyperplane. The linear SVM is required in linearly differentiating the information/data, that represents dividing the dataset into two different classes by a unique linear separation. Data is termed to as the linear differential and the classifier is written as:

$$Class1(Lowrisk) = (W * X + b) \geq 1, \forall X \qquad (1)$$

$$Class1(Highrisk) = (W * X + b) \leq -1, \forall X \qquad (2)$$

where, 'W' is a vector to Hyperplane, 'b' is a bias and 'x' is a matrix from the dataset.

The non-linear term itself referred to as the in deterministic data division, that represents a dataset that cannot be divided by the shortest route, that particular information is referred to as non-linear data and hence the classifier is written as:

$$K(X, Y) = (1 + X * Y)^d \qquad (3)$$

where, 'X' is data of low risk, 'Y' data of high risk and 'd' is degree of the polynomial. The RBF kernel represents a consequence that gives points

relies upon the measured interval from the initial origination or from any particular point is written as:

$$K(X, X1) = exp(-||X - X1||^2 \quad 2\sigma^2) \qquad (4)$$

where,$||X - X1||$ defines the distances between the two risk vectors and let $\gamma = 1/2\sigma^2$

$$K(X, X1) = exp(-\gamma||X - X1||^2) \qquad (5)$$

7.1.5 Gaussian Naive Bayes

When working with continuous data, an assumption often taken is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. The likelihood of the features is assumed to be-

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

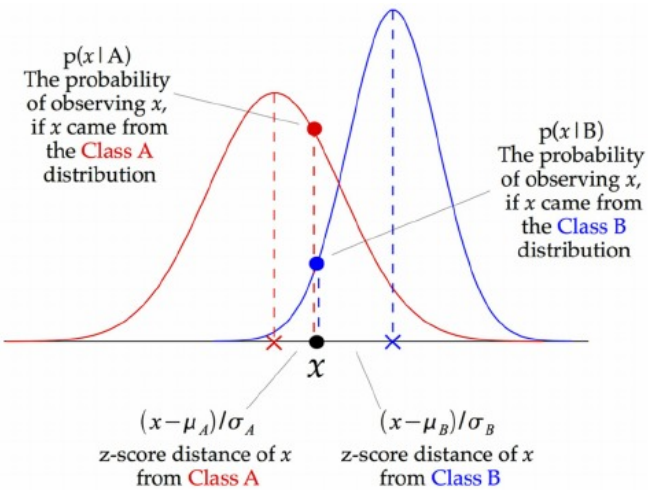
Sometimes assume variance

- is independent of Y (i.e.,σi),
- or independent of Xi (i.e., σk)
- or both(i.e, σ)

Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution.

An approach to create a simple model is to assume that the data is de-

scribed by a Gaussian distribution with no co-variance (independent dimensions) between dimensions. This model can be fit by simply finding the mean and standard deviation of the points within each label, which is all what is needed to define such a distribution.



The above illustration indicates how a Gaussian Naive Bayes (GNB) classifier works. At every data point, the z-score distance between that point and each class-mean is calculated, namely the distance from the class mean divided by the standard deviation of that class.

7.1.6 LightGBM (Light Gradient Boosting Machine)

LightGBM is a gradient boosting framework based on decision trees to increases the efficiency of the model and reduces memory usage.

It uses two novel techniques:

Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB) which fulfills the limitations of histogram-based algorithm that is primarily used in all GBDT (Gradient Boosting Decision Tree) frameworks. The two techniques of GOSS and EFB described below form the characteristics of

LightGBM Algorithm. They comprise together to make the model work efficiently and provide it a cutting edge over other GBDT frameworks.

1. Gradient-based One Side Sampling Technique for LightGBM:

   Different data instances have varied roles in the computation of information gain. The instances with larger gradients(i.e., under-trained instances) will contribute more to the information gain. GOSS keeps those instances with large gradients (e.g., larger than a predefined threshold, or among the top percentiles), and only randomly drop those instances with small gradients to retain the accuracy of information gain estimation. This treatment can lead to a more accurate gain estimation than uniformly random sampling, with the same target sampling rate, especially when the value of information gain has a large range.

2. Exclusive Feature Bundling Technique for LightGBM:

   High-dimensional data are usually very sparse which provides us a possibility of designing a nearly lossless approach to reduce the number of features. Specifically, in a sparse feature space, many features are mutually exclusive, i.e., they never take nonzero values simultaneously. The exclusive features can be safely bundled into a single feature (called an Exclusive Feature Bundle). Hence, the complexity of histogram building changes from O(#data $\times$ #feature) to O(#data $\times$ #bundle), while #bundle<<#feature . Hence, the speed for training framework is improved without hurting accuracy.

7.1.7 XGBoost

XGBoost is an open-source software library that implements optimized distributed gradient boosting machine learning algorithms under the Gradient Boosting framework.In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

The list of benefits and attributes of XGBoost is extensive, and includes the following:

- A large and growing list of data scientists globally that are actively contributing to XGBoost open source development
- Usage on a wide range of applications, including solving problems in regression, classification, ranking, and user-defined prediction challenges
- A library that's highly portable and currently runs on OS X, Windows, and Linux platforms
- Cloud integration that supports AWS, Azure, Yarn clusters, and other ecosystems
- Active production use in multiple organizations across various vertical market areas

- A library that was built from the ground up to be efficient, flexible, and portable

7.1.8 K-Nearest Neighbor

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data

Advantages of KNN Algorithm:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages of KNN Algorithm:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

7.1.9 Stochastic Gradient Descent (SGD)

The word 'stochastic' means a system or a process that is linked with a random probability. Hence, in Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. Stochastic Gradient Descent (SGD) is a simple yet efficient optimization algorithm used to find the values of parameters/coefficients of functions that minimize a cost function. In other words, it is used for discriminative learning of linear classifiers under convex loss functions such as SVM and Logistic regression. It has been successfully applied to large-scale datasets because the update to the coefficients is performed for each training instance, rather than at the end of instances.

SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than 10^5 training examples and more than 10^5

features.

The advantages of Stochastic Gradient Descent are:

- Efficiency
- Ease of implementation (lots of opportunities for code tuning).

The disadvantages of Stochastic Gradient Descent include:

- SGD requires a number of hyperparameters such as the regularization parameter and the number of iterations.
- SGD is sensitive to feature scaling.
- SGD is sensitive to feature scaling.

7.2 ASSET VISUALIZATION

At a facility level, technicians accessing the user-interface will not be trained in Artificial Intelligence and Big Data. The key considerations when defining this requirement are the visualization of machine behavior and the ability to depict the health of machinery or the entire facility, and take specific action as a result.

7.2.1 Non-Functional Requirement

Non-functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties.Non-functional requirements arise through the user needs, because of budget constraints, organizational policies, the need for inter-operability with other

software and hardware systems or because of external factors such as: -

1. Scalability

   Analytics platform must be applicable to a machine or facility of any size. The solution must be able to add assets without a need for any incremental investment in hardware, software or dedicated labor hours.

2. Performance

   The objective for an industrial analytics platform is to provide the production facility with accurate and timely data.

3. Requirements

   There must be a user interface to configure the network. There must be an option for the user to select An option to view the performance parameters. The system should be user friendly, so that the client application is available at the system tray and the user has to just click to select any options.

4. Response time

   Response time is the elapsed time between an inquiry on a system and the response to that inquiry. Used as a measurement of system performance, response time may refer to service requests in a variety of technologies. Low response times may be critical to successful computing.

5. Maintainability

   Maintainability is an important quality attribute and a difficult concept as it involves a number of measurements. Quality estimation means estimating maintainability of software. Maintainability is a set of attributes

that bear on the effort needed to make specified modification

6. Usability

   One problem facing designers of interactive systems is catering to the wide range of users who will use a particular application. Understanding the user is critical to designing a usable interface. There are a number of ways of addressing this problem,including improved design methodologies using "intuitive" interface styles, adaptive interfaces, and better training and user support materials.

8 SYSTEM DEVELOPMENT METHODOLOGY

The methodology of software development is the method in managing project development. There are many models of the methodology available such as Waterfall model model, Incremental model, RAD model, Agile model, Iterative model and Spiral model. However, it still needs to be considered by the developer to decide which will be used in the project. The methodology model is useful to manage the project efficiently and able to help developers from getting any problem during time of development. Also, it helps to achieve the objective and scope of the projects. In order to build the project, it needs to understand the stakeholder requirements.

Methodology provides a framework for undertaking the proposed DM modeling. The methodology is a system comprising steps that transform raw data into recognized data patterns to extract knowledge for users.

The development method followed in this project is the 'waterfall model'.

9 MODEL PHASES

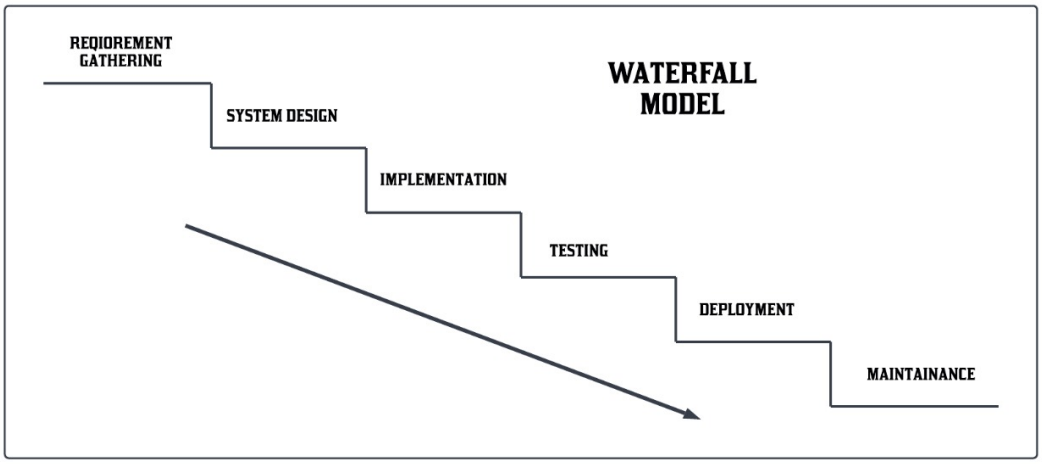


Figure 3: waterfall

Initially the requirement gathering is the phase where the title of the project has been discussed. From that discussion, the Stroke Prediction System using ML Algorithms has been proposed. The requirements and the risks were assessed after doing thorough study on the existing systems and doing literature review about other existing research. Next system design is the phase where system specifications are translated into a software representation using the requirements. In this phase the designer emphasizes algorithms, data structures, software architecture, etc. The implementation phase involves the programmers to start coding in order to give a full sketch of the product. That is, the system specifications are only converted into machine readable compute code. The output of this phase is typically the library, executables, user manuals and additional software documentation. Next in the testing phase, all programs and models are integrated and tested to ensure that the complete system meets the software requirements. The testing is concerned with verification and validation. Finally, the maintenance phase a.k.a the longest phase, where the software is updated to fulfill

the changing customer need, adapt to accommodate change in the external environment, correct errors and oversights previously undetected in the testing phase, and enhance the efficiency of the software.

10 DESIGN USING UML

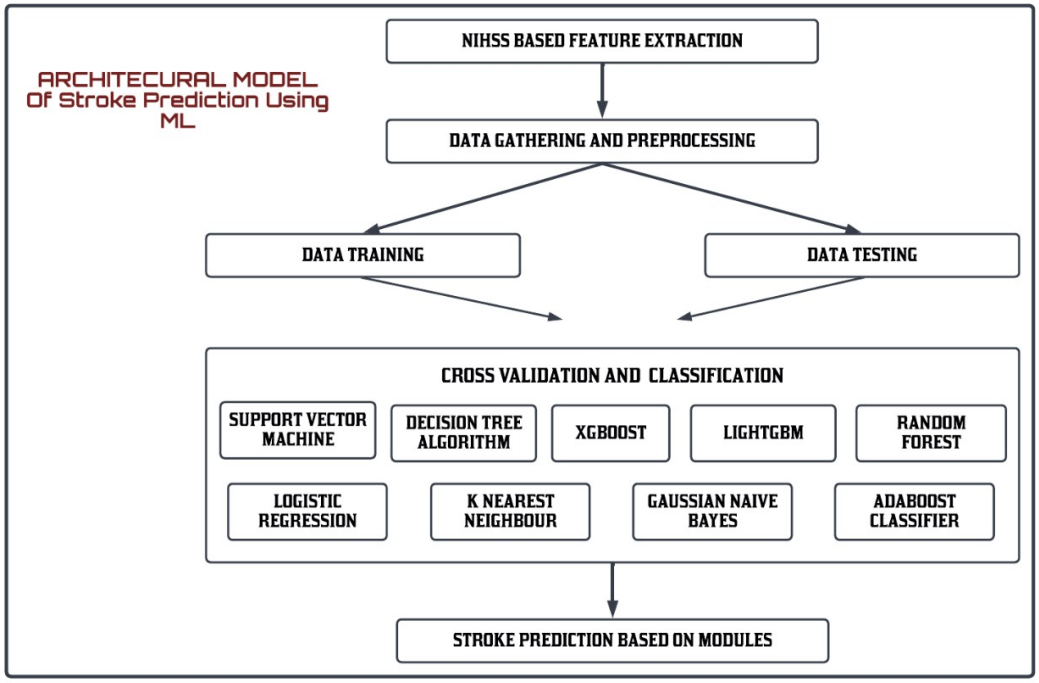


Figure 4: Architectural

UML diagrams specify the process of communication of the system along with collaborating objects within the process using both static as well as dynamic UML diagrams. Since the development of Object-Oriented applications, it has been getting difficult to develop and manage high quality applications in a reasonable time-frame.

As a result of this challenge and the need for a universal object modeling language every one could use, the Unified Modeling Language (UML) is the Information industries version of blue print. It is a method for describing the systems architecture in detail. Easier to build or maintain a system, and to ensure that the system will hold up to the requirement changes.

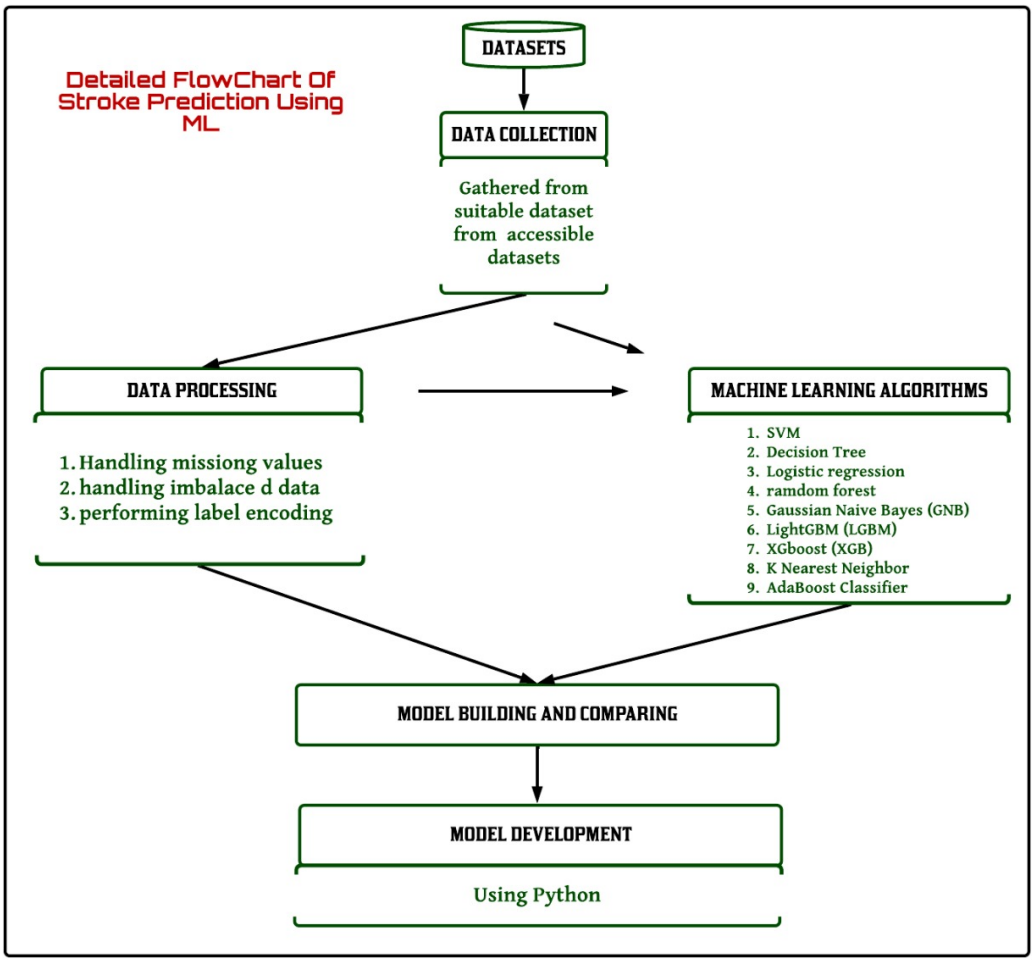


Figure 5: Flow chart

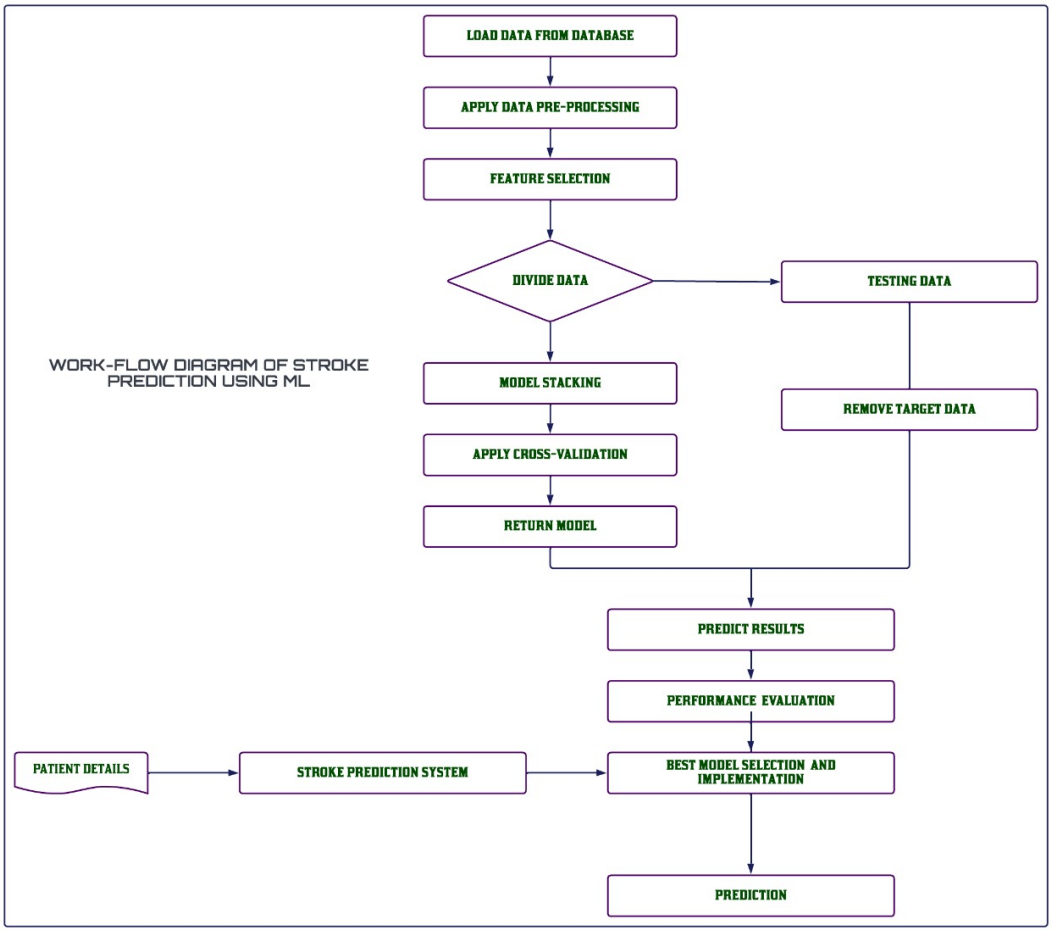


Figure 6: Work flow

11 TOOLS

For application development, the following Software Requirements are

Operating System: Windows 7 , 10 or 11.

Language: python

Tools: Jupyter notebook, Microsoft Excel (Optional).

Technologies used:

12 SOFTWARE REQUIREMENT

Operating System: Any OS with clients to access the internet

Network: Wi-Fi Internet or cellular Network

Lucidchart, Draw.io: Create and design UML Diagrams

Github Versioning Control

Google Chrome Medium to find references to do system testing, display and run application

13 HARDWARE REQUIREMENT

For application development, the following Software Requirements are:

Processor: Intel or high RAM: 1024 MB

Space on disk: minimum 100mb

For running the application:

Device: Any device that can access the internet

Minimum space to execute: 20 MB

The effectiveness of the proposal is evaluated by conducting experiments with a cluster formed by 3 nodes with identical setting, configured with an Intel CORE™ i7-10750H processor (2.60GHZ, 4 Cores, 16GB RAM, running

Windows 11 Home Single Language with 64-bit operating system, x64-based processor)

14 Implementation

The implementation phase of the project is where the detailed design is actually transformed into working code. Aim of the phase is to translate the design into the best possible solution in a suitable programming language. This part covers the implementation aspects of the project, giving details of the programming language and development environment used. It also gives an overview of the core modules of the project with their step by step flow. The implementation stage requires the following tasks.

1. Careful planning
2. Investigation of the system and constraints.
3. Design of methods to achieve the changeover.
4. Evaluation of the changeover method.
5. Correct decisions regarding selection of the platform
6. Appropriate selection of the language for application development

```
# import package
# open dataset
filename = "/content/drive/MyDrive/data/train_2v.csv"
data = pd.read_csv(filename)
with pd.option_context('expand_frame_repr', False):
    print(data.head())
print("Data shape: {}".format(data.shape))
```

```
      id  gender   age  hypertension  heart_disease ever_married      work_type Residence_type  avg_glucose_level   bmi   smoking_status  stroke
0  30669    Male   3.0             0              0           No       children          Rural              95.12  18.0              NaN       0
1  30468    Male  58.0             1              0          Yes        Private          Urban              87.96  39.2     never smoked       0
2  16523  Female   8.0             0              0           No        Private          Urban             110.89  17.6              NaN       0
3  56543  Female  70.0             0              0          Yes        Private          Rural              69.04  35.9  formerly smoked       0
4  46136    Male  14.0             0              0           No  Never_worked          Rural             161.28  19.1              NaN       0
Data shape: (43400, 12)
```

Figure 7: Dataset loaded

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43400 entries, 0 to 43399
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 43400 non-null  int64
 1   gender             43400 non-null  object
 2   age                43400 non-null  float64
 3   hypertension       43400 non-null  int64
 4   heart_disease      43400 non-null  int64
 5   ever_married       43400 non-null  object
 6   work_type          43400 non-null  object
 7   Residence_type     43400 non-null  object
 8   avg_glucose_level  43400 non-null  float64
 9   bmi                41938 non-null  float64
 10  smoking_status     30108 non-null  object
 11  stroke             43400 non-null  int64
dtypes: float64(3), int64(4), object(5)
memory usage: 4.0+ MB
```

Figure 8: Data info

15 FUTURE SCOPE

The early prognosis of stroke can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. This project helps to predict stroke risk using prediction models in older people and for people who are addicted to the risk factors as mentioned in the project.

This system is ML based, user-friendly, scalable, reliable and an expandable system. We will be using (requirements mention karna hai) The proposed working model can also help in reducing treatment costs by providing Initial diagnostics in time

In future, the same project can be extended to give the stroke percentage using the output of the current project. This project can also be used to find the stroke probabilities in young people and underage people by collecting respective risk factor information and doctors consulting. As we have developed a generalized system, in future we can use this system for the analysis of different data sets. There are many possible improvements that could be explored to improve the scalability and accuracy of this prediction system.

References

[1] "Concept of stroke by healthline," [Online]. Available:
https://www.cdc.gov/stroke/index.htm.

[2] "Statistics of stroke by Centers for disease control and prevention," [Online]. Available:
https://www.cdc.gov/stroke/facts.htm.

[3] 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

[4] 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

[5] 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

[6] Xie, Y., Jiang, B., Gong, E., Li, Y., Zhu, G., Michel, P., 760 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

[7] 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

[8] 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

[9] 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.

[10] https://github.com/eddieir/HealthCare/blob/master/heartstroke/input/train2v.csv (dataset used by us)

[11] "Development of an Algorithm for Stroke Prediction: A National Health Insurance Database Study" - Min SN, Park SJ, Kim DJ, Subramaniyam M, Lee KS

[12] "Effective Analysis and Predictive Model of Stroke Disease using Clas-

sification Methods" - A.Sudha, P.Gayathri, N.Jaisankar

[13] "Focus on stroke: Predicting and preventing stroke" - Michael Regnier

[14] 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 & Applications, vol. 32, no. 3, pp. 817–828, 2020.