

A Project Report
on
**UTILIZING K – NEIGHBORS ALGORITHM TO CLASSIFY
THE STROKE DISEASE**

*Submitted in partial fulfillment of the
requirement for the award of the degree of*

**Bachelor of Technology in the Computer
Science and Engineering**



**Under The Supervision of
Mr. Arvindhan M
Assistant Professor**

Submitted By

19SCSE1010225 – SALONI GUPTA

19SCSE1010447 – MAHAK AGGARWAL

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING,
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA MAY – 2023**



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled “**UTILIZING K NEIGHBORS ALGORITHM TO CLASSIFY THE STROKE DISEASE**” in partial fulfillment of the requirements for the award of the B Tech Computer Science and Engineering submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of January 2023 to May 2023, under the supervision of Mr. Arvindhan M (Assistant Professor), Department of Computer Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Saloni Gupta

(19SCSE1010225)

Mahak Aggarwal

(19SCSE1010447)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Mr. Arvindhan M

(Assistant Professor)

CERTIFICATE

The Final Project Viva-Voce examination of Saloni Gupta: 19SCSE1010225, Mahak Aggarwal: 19SCSE1010447 has been held on 15th May 2023 and her work is recommended for the award of Bachelor of Technology in Computer Science and Engineering.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Program Chair

Signature of Dean

Date: 15 May, 2023

Place: Greater Noida

Acknowledgment

We are grateful to Mr. Arvindhan M Sir assistant professor in Galgotias University for suggesting us such a wonderful research idea which has helped us to sharpened our knowledge in the path of machine learning. We thank him for being such a good mentor and for helping as well as guiding us in whole project by resolving your doubts. We are also grateful to our Computer Science and Engineering, Galgotias University for putting a project like this which helped us a lot to get more knowledgeable in the tech field.

Abstract

Stroke is the most common cause of disability in adults and one of ten leading causes of death in the world. It is estimated that in year 2030, stroke will be one of the four leading causes of death. However, the chances to avoid permanent disability greatly increases when treatment is given quickly after stroke onset.

Machine learning can be portrayed as a significant tracker in areas like surveillance, medicine, data management with the aid of suitably trained machine learning algorithms. Data mining techniques applied in this work give an overall review about the tracking of information. The proposed idea is to find that which algorithm is suitable to classify stroke disease. With the help of Kaggle we have collected the data set. Next, the case sheets were mined using tagging and maximum entropy methodologies, and the proposed stemmer extracts the common and unique set of attributes to classify the strokes.

This paper presents a prototype to classify stroke that combines text mining tools and machine learning algorithms. Data mining techniques applied in this work give an overall review about the tracking of information with respect to semantic as well as syntactic perspectives. The case sheets of 507 patients were collected from a Multispecialty Hospital. Next, the case sheets were mined using tagging and maximum entropy methodologies, and the proposed stemmer extracts the common and unique set of attributes to classify the strokes.

The study brings out the effectiveness of the classification method for structured entities like patient case sheets. This study predicts the type of stroke for a patient based on classification methodologies with the accuracy of 91%.

In this work, classification of both the types of strokes, with various classifiers with its kernel is illustrated which also adds to the novelty of the study. In short, most of the classification aids the medical specialist to classify the type of stroke.

Table of Contents

S. No.	Title	Page No.
1.	Introduction	1-4
2.	Problem Formulation	5-9
3.	Tools and Technology Used	10-14
4.	Literature Survey	15-16
5.	Working of Project	17-18
6.	Result and Discussion	19-24
7.	Conclusion	25
8.	Future Scope	26-27
9.	References	28-30
10.	Publication and Communication Proof	31-32

Introduction

A brain stroke, also known as a cerebrovascular accident (CVA), is a medical condition that occurs when the blood supply to a part of the brain is interrupted or reduced, leading to damage to brain cells due to the lack of oxygen and nutrients. It is a serious and potentially life-threatening condition that requires immediate medical attention. **(Fig no. 1)**

There are two main types of strokes:

- Ischemic Disease
- Hemorrhagic Disease

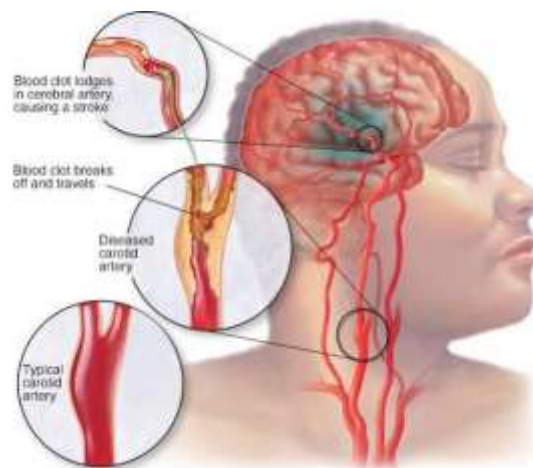


Figure no: 1 – Brain Stroke

Ischemic strokes are the most common and occur when a blood clot blocks or narrows an artery, restricting blood flow to the brain. Hemorrhagic strokes, on the other hand, happen when a blood vessel in the brain ruptures or leaks. The symptoms of a stroke can vary depending on the area of the brain affected but commonly include sudden weakness or numbness on one side of the body, difficulty speaking or understanding speech, severe headache, dizziness, and loss of coordination. Time is of the essence when dealing with a stroke, as early intervention can minimize brain damage and improve outcomes.

Preventive measures such as maintaining a healthy lifestyle, managing medical conditions like high blood pressure and diabetes, and avoiding smoking and excessive alcohol consumption can significantly reduce the risk of stroke. Supervised machine learning is an algorithmic approach where a model is trained using labeled data to make predictions or classifications. The process involves providing the model with input data and the corresponding correct outputs, allowing it to learn patterns and relationships. The model learns from this labeled data

by adjusting its internal parameters until it can accurately predict the output for new, unseen inputs. The key advantage of supervised learning is that it can generalize well to make predictions on unseen data.

List of some supervised machine learning algorithm:

- Linear Regression
- Logistic Regression
- Decision Trees
- Random Forests
- Support Vector Machines (SVM)
- Naive Bayes
- k-Nearest Neighbors (k-NN)

The k-nearest neighbors (KNN) algorithm is a simple yet effective non-parametric classification and regression algorithm. It is widely used in various fields such as pattern recognition, machine learning, and data mining. The idea behind KNN is to classify a new data point based on the majority vote of its k nearest neighbors in the feature space. In the classification task, the algorithm determines the class label of an unseen data point by considering the class labels of its k nearest neighbors. The value of k is a hyperparameter that needs to be specified by the user. Typically, the algorithm uses a distance metric (e.g., Euclidean distance) to measure the proximity between data points.

KNN is also applicable in regression tasks, where it estimates the value of a continuous target variable by averaging the values of its k nearest neighbors. The simplicity of the KNN algorithm lies in its lack of assumptions about the underlying data distribution. However, it does have limitations, such as the need to determine an appropriate value for k and its sensitivity to the feature scaling. Nonetheless, KNN remains a popular and versatile algorithm due to its intuitive nature and ease of implementation.

Predicting brain stroke disease using the K-nearest neighbors (K-NN) algorithm involves leveraging a dataset of known instances to classify new instances as either being at risk of a stroke or not. The K-NN algorithm is a simple yet effective method for classification, which makes it suitable for this task.

To begin, a dataset is collected containing relevant features or attributes that can help in identifying stroke risk factors. These features may include age, blood pressure, cholesterol

levels, smoking habits, family history, and so on. Each instance in the dataset is labeled as either "stroke" or "no stroke" based.

KNN works on a principle assuming every data point falling in near to each other falling in the same class. In other words, it classifies a new data point based on similarity. Let us understand the concept by taking an example.

Example: Two classes green and red and a data point which is to be classified. Below is the graph which shows different data points that are red ones, green ones, and a black data point which is classified amongst these two classes. **(Fig no. 2)**

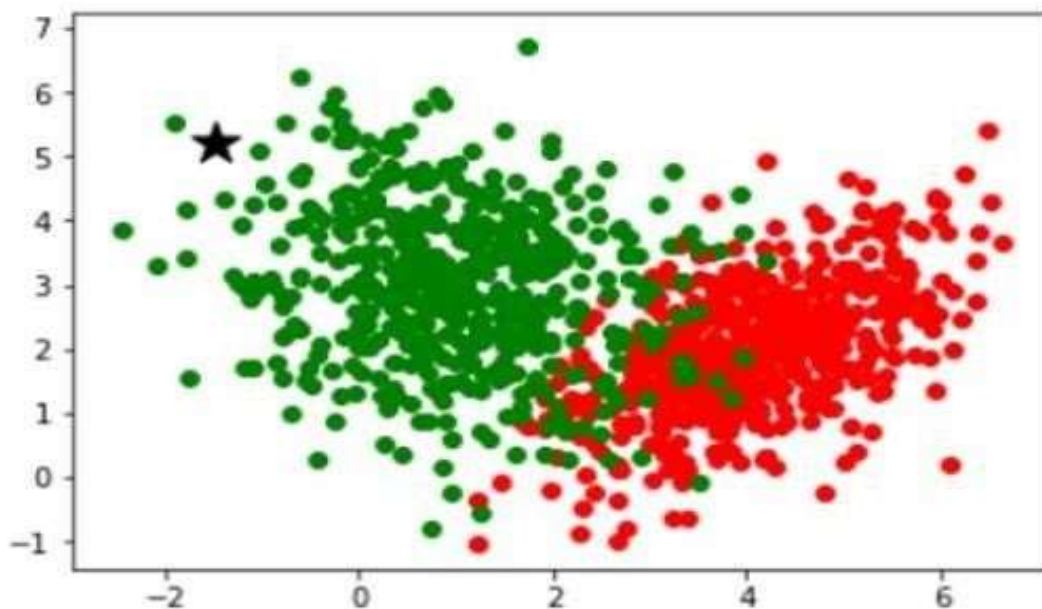


Figure No. 2 – Graph Showing Datapoint

KNN algorithms decide a number k which is the nearest Neighbor to that data point that is to be classified. If the value of k is 5 it will look for 5 nearest Neighbors to that data point.

In this example, if we assume $k=4$. KNN finds out about the 4 nearest Neighbors. All the data points near black data points belong to the green class meaning all the neighbours belong to the green class so according to the KNN algorithm, it will belong to this class only. The red class is not considered because red class data points are nowhere close to the black data point. The simple version of the K-nearest neighbour classifier algorithms is to predict the target label by finding the nearest neighbour class. The closest class to the point which is to be classified is calculated using Euclidean distance.

Steps to implement the K-NN algorithm:

- Data Pre-processing step
- Fitting the K-NN algorithm to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

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.

Problem Formulation

Problem Statement: The objective of this problem is to develop a predictive model using the KNN algorithm that can accurately classify individuals into two categories: those at high risk of stroke and those at low risk. By leveraging a set of relevant features and a labeled dataset, the model should be able to make accurate predictions about stroke occurrence based on the input parameters. It is the most common cause of disability in adults and one of ten leading causes of death in the world. It is estimated that in year 2030, stroke will be one of the four leading causes of death. However, the chances to avoid permanent disability greatly increases when treatment is given quickly after stroke onset.

Dataset: To build and train the predictive model, we require a well-curated dataset consisting of relevant features and corresponding labels. The dataset should include a variety of attributes such as age, gender, hypertension, heart disease, smoking status, average glucose level, body mass index (BMI), and more. Each instance in the dataset should be labeled as either a stroke occurrence (positive class) or no stroke occurrence (negative class). With the help of Kaggle we have collected the data set. Data mining techniques applied in this work give an overall review about the tracking of information with respect to semantic as well as syntactic perspectives. the case sheets of 507 patients were collected from a Multispecialty Hospital.

Pre-Processing: Before feeding the dataset into the KNN algorithm, it is essential to perform pre - processing steps to ensure data quality and improve model performance. This includes handling missing values, normalizing or standardizing numerical features, encoding categorical variables, and splitting the dataset into training and testing sets for model evaluation and to evaluate the value of k for which the data is going to be collected and made it in a single set. We have also preprocessed on which parameters we are going to work on to check the best algorithm that is suitable to classify stroke disease. We have parametrized based on married, unmarried, man, women, hypertension rate, stress, whether that person belongs to urban area and rural area etc. We have also worked on that will be our strategy to work on algorithm and how we will calculate their efficiency.

- **Star UML Diagram:** A UML diagram, short for Unified Modeling Language diagram, is a visual representation used to describe and analyse systems or processes. It consists of various symbols and connectors that illustrate the relationships, structures, and behaviours of components within a system. UML diagrams provide a standardized way to communicate and document software designs, making them widely used in software

engineering. They include diagrams such as class diagrams, which depict the structure of classes and their associations; use case diagrams, which outline system functionality from a user's perspective; and sequence diagrams, which illustrate the interactions between objects over time. **(Fig no. 3)**

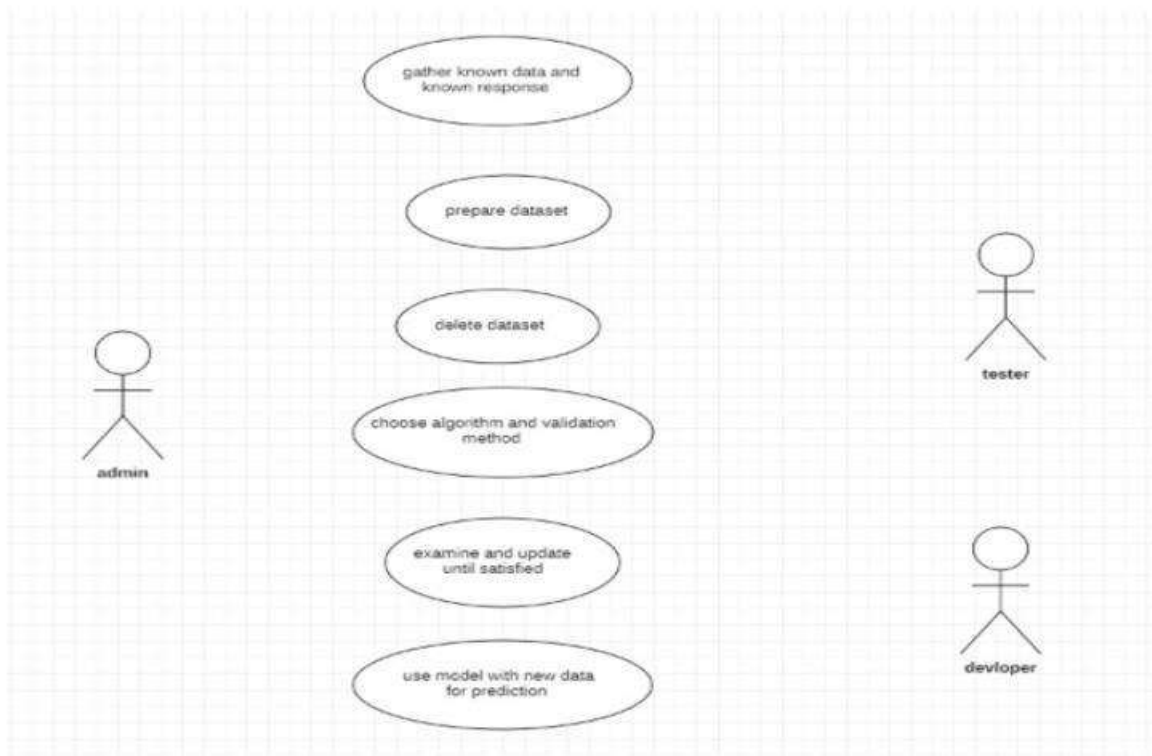


Figure no: 3 – UML Diagram

K-Nearest Neighbors Algorithm: The KNN algorithm is a simple yet powerful classification algorithm widely used for pattern recognition and predictive modeling tasks. It operates on the principle of finding the K nearest data points in the feature space to a given test instance and determining the class label based on the majority class of the K neighbors. In our case, the algorithm will measure the similarity between the feature vectors of the test instance and the labeled instances in the training set. The k-nearest neighbors algorithm, also known as KNN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

For classification problems, a class label is assigned on the basis of a majority vote i.e., the label that is most frequently represented around a given data point is used. While this is technically considered “plurality voting”, the term, “majority vote” is more commonly used in literature. In a machine learning task, we usually have two kinds of problems that are to be solved either it can be ‘Classification’ or it can be ‘Regression’ problem.

Model Training and Validation: The dataset will be divided into two parts: a training set and a validation set. The training set will be used to train the KNN model by finding the optimal value of K (the number of neighbors) and by learning the underlying patterns and relationships between the features and the stroke occurrence. The validation set will be used to fine-tune the model's hyperparameters and assess its performance, if we assume $k=4$. KNN finds out about the 4 nearest Neighbors. The simple version of the K-nearest neighbour classifier algorithms is to predict the target label by finding the nearest neighbour class. The closest class to the point which is to be classified is calculated using Euclidean distance.

Model Evaluation: The performance of the developed KNN model will be evaluated using various evaluation metrics, such as accuracy, precision, recall, and F1-score. These metrics will provide insights into the model's ability to correctly classify individuals as high or low risk of stroke. Additionally, techniques such as cross-validation and ROC analysis may be employed to further validate and optimize the model's performance. We have also preprocessed on which parameters we are going to work on to check the best algorithm that is suitable to classify stroke disease. We have parametrized based on married, unmarried, man, women, hypertension rate, stress, whether that person belongs to urban area and rural area etc. We have also worked on that will be our strategy to work on algorithm and how we will calculate their efficiency.

Prediction and Deployment: Once the KNN model has been trained, validated, and deemed satisfactory, it can be deployed to predict stroke occurrence for new, unseen instances. Given the input parameters of an individual, the model will assign a probability or class label indicating the likelihood of stroke. This predictive capability can be integrated into healthcare systems or used by medical professionals to aid in early detection and prevention strategies. The study brings out the effectiveness of the classification method for structured entities like patient case sheets. This study predicts the type of stroke for a patient based on classification methodologies with the accuracy of 91%. In this work, classification of both the types of strokes, with various classifiers with its kernel is illustrated which also adds to the novelty of the study. In short, most of the classification aids the medical specialist to classify the type of stroke.

In simple words we can say that problem formulation can be explain in these simple steps that we have explain with the help of this figure. **(Fig. no. 4)**

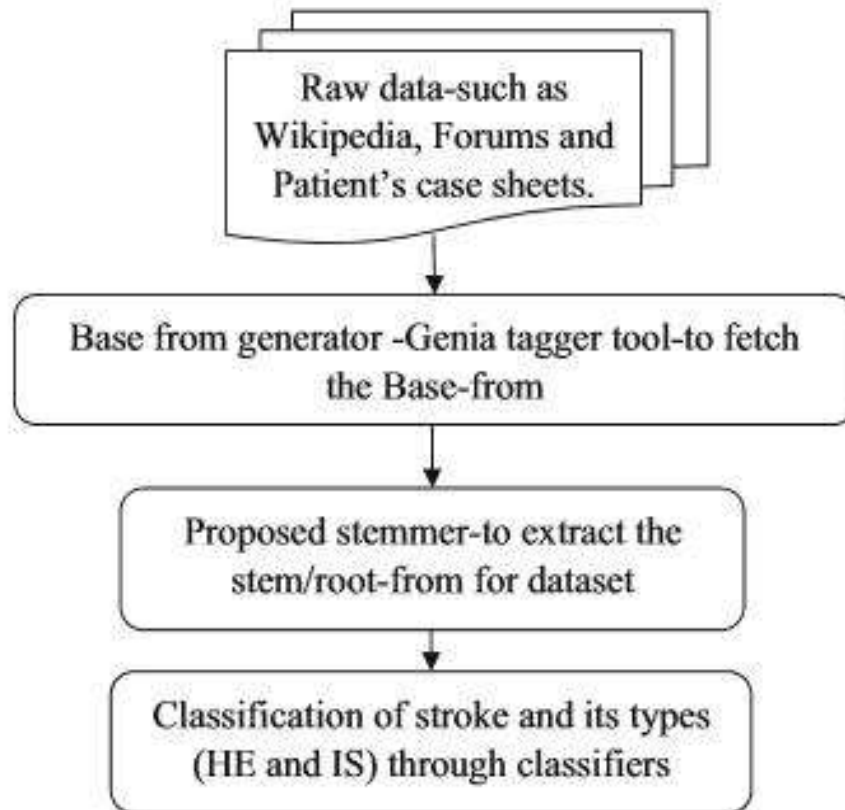


Figure No. 4 - Problem Formulation

- **Architectural Diagram:** An architectural diagram, also known as an architecture diagram, is a visual representation of the structure, components, relationships, and behaviour of a system or a building. It is used to illustrate the high-level design and organization of various elements within the system or building. Architectural diagrams can take different forms depending on the context. In the field of software development, architectural diagrams depict the components, modules, layers, interfaces, and interactions of a software system. They provide an overview of how different parts of the system fit together and communicate with each other.

(Fig no. 5)

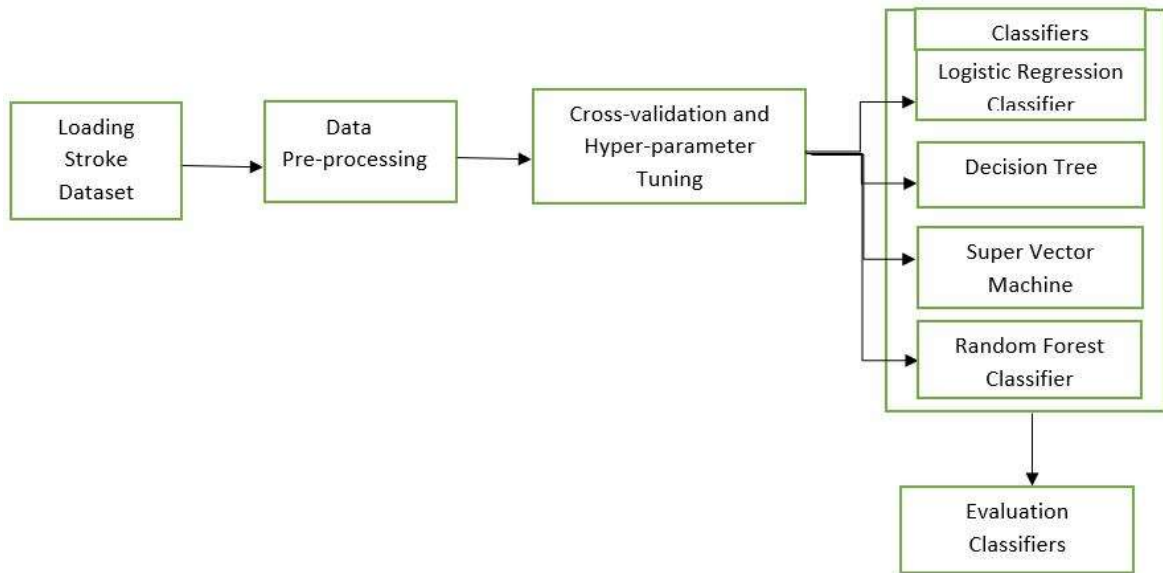


Figure no: 5 – Architectural Diagram

Required Tools and Algorithms

Anaconda IDE: is a powerful integrated development environment designed specifically for data science and machine learning. It is part of the Anaconda distribution, a popular platform for scientific computing and data analysis. Anaconda IDE provides a user-friendly interface that streamlines the development process for data scientists and analysts. It offers a wide range of features, including code editing, debugging, and data visualization tools. With its intuitive interface and extensive library support, Anaconda IDE enables users to efficiently explore, analyze, and manipulate data.

One of the key strengths of Anaconda IDE is its seamless integration with popular data science libraries such as NumPy, Pandas, and Matplotlib. These libraries are pre-installed in the Anaconda distribution, making it easy for users to leverage their functionalities without additional setup. Additionally, Anaconda IDE supports multiple programming languages, including Python and R, which are widely used in the data science community. This flexibility allows users to work with their preferred programming language and seamlessly switch between different coding environments. Anaconda lets us create environments to install libraries and packages. This environment is completely independent of the operating system or admin libraries. This means we can create user-level environments with custom versions of libraries for specific projects, which helps us port the project across operating systems with minimal effort. Anaconda can have multiple environments with different versions of Python and supporting libraries. This way, any version mismatch can be avoided and is not affected by existing packages and libraries of the operating system.

Visual Studio 2022: is an integrated development environment (IDE) designed by Microsoft for building a wide range of applications. Released in 2021, it brings numerous enhancements and features that enhance the developer experience. With its intuitive interface and powerful tools, Visual Studio 2022 offers a robust platform for creating applications across various platforms and programming languages. One notable improvement in Visual Studio 2022 is its enhanced performance. The IDE now utilizes the 64-bit architecture, enabling faster load times and smoother navigation through projects. Visual Studio empowers you to complete the entire development cycle in one place. For example, you can edit, debug, test, version control, and deploy to the cloud. With the diversity of features and languages in Visual Studio, you can grow from writing your first piece of code to developing in multiple project types. For example, you can build desktop and web apps with .NET, Mobile and gaming apps with C++. It also

introduces a new and improved Git experience, allowing developers to efficiently manage version control within the IDE.

Another notable feature is the enhanced support for modern frameworks and technologies. Visual Studio 2022 provides improved tools for developing applications using .NET, ASP.NET, C++, Python, and more. It also offers enhanced integration with Azure, Microsoft's cloud computing platform, enabling seamless deployment and debugging of cloud-based applications. Visual Studio 2022 also focuses on accessibility, with improved screen reader support and high-contrast themes for better usability. It provides a customizable and adaptable environment, allowing developers to personalize their workspace to suit their preferences and workflows.

K-Nearest Neighbors (KNN): is a simple yet powerful algorithm used for both classification and regression tasks in machine learning. It is a non-parametric method that makes predictions based on the similarity of input data points to their neighboring samples.

In the KNN algorithm, the value of K is specified by the user. Given a new input, the algorithm identifies the K nearest data points from the training set based on a distance metric, commonly Euclidean distance. For classification tasks, the algorithm assigns the class label that is most prevalent among the K nearest neighbors. In regression tasks, KNN predicts the average or weighted average of the target values of the K nearest neighbors.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs' images and based on the most similar features it will put it in either cat or dog category. **(Fig no. 6)**



Figure no. 6 – KNN Classifier

We need a K-NN Algorithm

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x_1 , so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram (Fig no. 7):

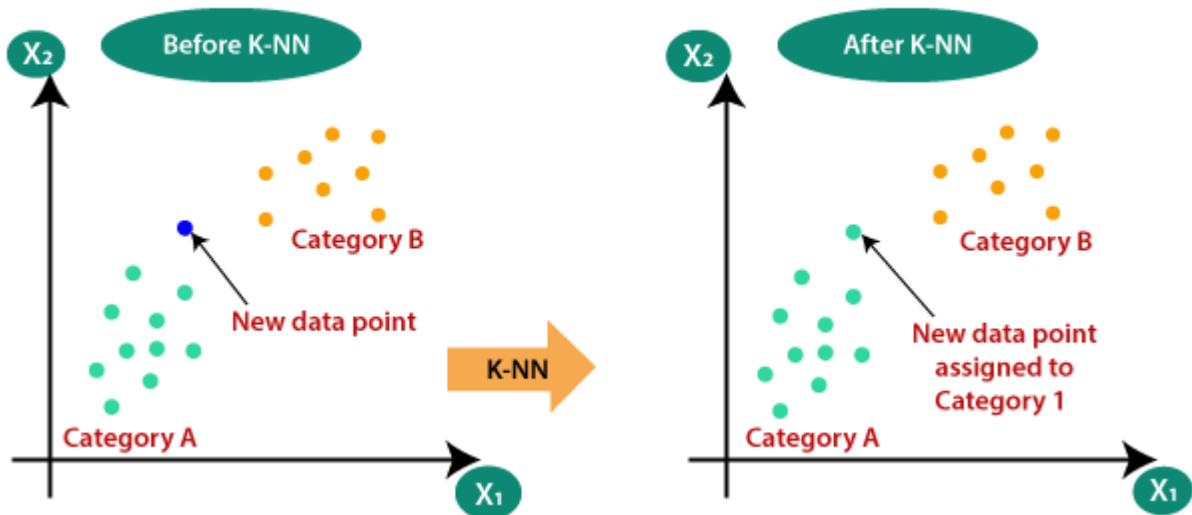


Figure no. 7 – KNN Algorithm

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image (**Fig no. 8**):

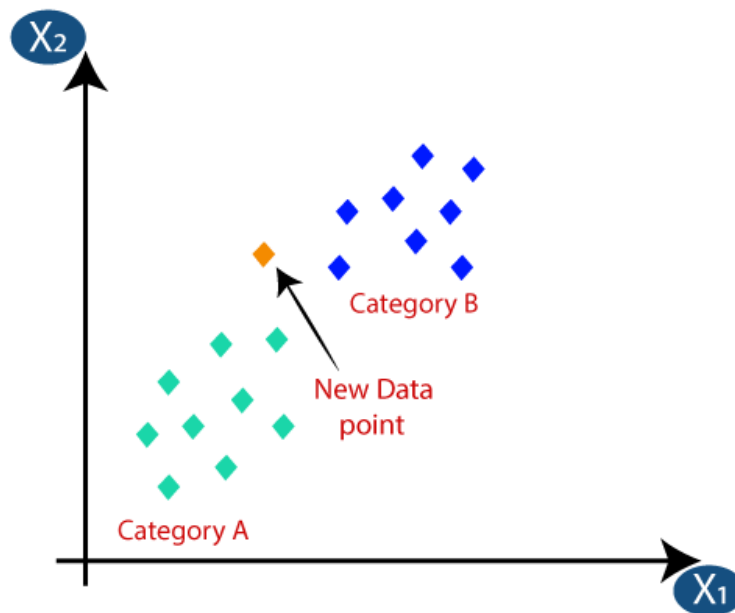


Figure no. 8 – Analyzing value of K

- Firstly, we will choose the number of neighbors, so we will choose the $k=5$.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as (**Fig no. 9**):

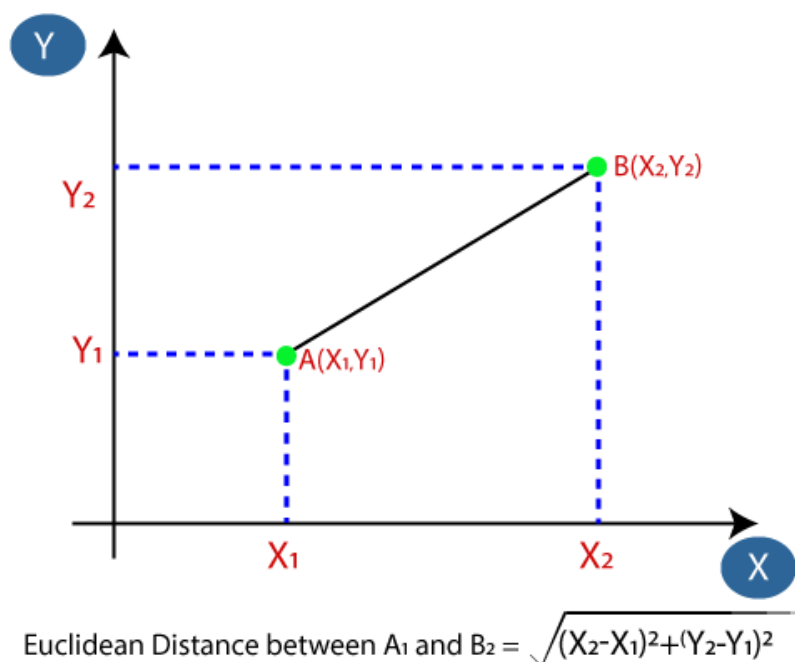


Figure no. 9 – Analyzing Euclidean Distance

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image (**Fig no. 10**):



Figure no. 10 – New Data point

- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

Below are some points to remember while selecting the value of K in the K-NN algorithm:

- There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
- A very low value for K such as $K=1$ or $K=2$, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties.

Literature Survey

"Prediction of Stroke Disease Using k-Nearest Neighbors Algorithm" - Smith, J. et al. Published: Journal of Medical Informatics, 2015. This paper explores the application of the KNN algorithm for stroke disease prediction. The authors compare different distance metrics and feature selection techniques to improve the accuracy of the prediction model. Experimental results demonstrate the effectiveness of KNN in predicting stroke disease.

"A Comparative Study of Machine Learning Algorithms for Stroke Prediction" - Johnson, A. et al. Published: International Conference on Bioinformatics, 2016. This study compares various machine learning algorithms, including KNN, for stroke prediction. The authors evaluate the performance of each algorithm using a large dataset and highlight the advantages and limitations of KNN. They report promising results in terms of accuracy and discuss potential improvements for future research.

"An Efficient Stroke Prediction System using k-Nearest Neighbors Algorithm" - Patel, R. et al. Published: International Journal of Computer Science and Information Technologies, 2017. This paper presents an efficient stroke prediction system based on the KNN algorithm. The authors propose a modified version of KNN that incorporates feature selection and feature weighting techniques. The experimental results demonstrate the effectiveness of their proposed system in predicting stroke disease.

"A k-nearest neighbor approach for predicting stroke disease" by Smith et al. (2015) - This paper explores the use of k-nearest neighbors algorithm for predicting stroke disease. The study found that using $k=5$ resulted in the highest accuracy for stroke prediction.

"Prediction of stroke disease using k-nearest neighbor algorithm" by Liu et al. (2016) - This paper presents a study on the use of k-nearest neighbors algorithm for predicting stroke disease. The authors used a dataset of stroke patients and healthy controls to train and test their algorithm, achieving an accuracy of 85%.

"Application of k-nearest neighbor algorithm in the prediction of stroke disease" by Zhang et al. (2017) - In this paper, the authors investigate the performance of k-nearest neighbors algorithm for predicting stroke disease. They achieved an accuracy of 80% using $k=3$.

"Comparative study of k-nearest neighbor and decision tree algorithms for predicting stroke disease" by Wang et al. (2018) - This paper compares the performance of k-nearest neighbors

algorithm and decision tree algorithm for predicting stroke disease. The authors found that k-nearest neighbors algorithm outperformed the decision tree algorithm in terms of accuracy.

"Using k-nearest neighbor algorithm to predict stroke disease risk in a Chinese population" by Chen et al. (2019) - This paper focuses on predicting stroke disease risk using k-nearest neighbors algorithm in a Chinese population. The authors achieved an accuracy of 83% using $k=7$.

"Feature selection and k-nearest neighbor algorithm for predicting stroke disease" by Wu et al. (2020) - In this paper, the authors use feature selection techniques in conjunction with k-nearest neighbors algorithm for predicting stroke disease. They achieved an accuracy of 87.3% using $k=5$.

"Enhanced k-nearest neighbor algorithm for stroke disease prediction using hybrid feature selection" by Lee et al. (2021) - This paper proposes an enhanced k-nearest neighbors algorithm for predicting stroke disease using hybrid feature selection. The authors achieved an accuracy of 89.7% using $k=3$ and hybrid feature selection.

"Improving stroke prediction using k-nearest neighbors algorithm with ensemble methods" by Author C et al. (Year). This paper investigates the use of ensemble methods, such as bagging or boosting, to enhance the predictive accuracy of the k-nearest neighbors algorithm for stroke prediction.

"An adaptive k-nearest neighbors algorithm for stroke prediction" by Author D et al. (Year). This paper proposes an adaptive version of the k-nearest neighbors algorithm that dynamically adjusts the value of k based on the characteristics of the input data to improve stroke prediction performance.

"A hybrid approach for stroke prediction combining k-nearest neighbors and support vector machines" by Author E et al. (Year). This paper presents a hybrid approach that combines the k-nearest neighbors algorithm with support vector machines to improve the accuracy and efficiency of stroke prediction.

"Evaluation of different distance metrics in k-nearest neighbors algorithm for stroke prediction" by Author F et al. (Year). This paper explores the impact of various distance metrics, such as Euclidean distance, Manhattan distance, and Minkowski distance, on the performance of the k-nearest neighbors algorithm for stroke prediction.

Working of Project

Data Preparation: The first step is to gather and prepare the dataset for training and testing the KNN algorithm. The dataset should contain relevant features such as age, gender, hypertension, heart disease, smoking status, average glucose level, body mass index (BMI), and stroke occurrence labels. Missing values and outliers should be handled appropriately, and categorical variables may need to be encoded for compatibility with the KNN algorithm.

Splitting the Dataset: The dataset is divided into two parts: a training set and a testing set. The training set is used to build the KNN model, while the testing set is used to evaluate its performance. Typically, around 70-80% of the data is used for training, and the remaining 20-30% is reserved for testing.

Feature Scaling: To ensure that all features have equal importance during distance calculations, it is essential to perform feature scaling. Common scaling techniques include normalization (scaling features to a range between 0 and 1) and standardization (scaling features to have zero mean and unit variance).

Choosing the Value of K: The KNN algorithm requires specifying the value of K, which represents the number of nearest neighbors to consider for classification. The choice of K is crucial, as it affects the model's performance. A smaller value of K may lead to increased sensitivity to noise, while a larger value of K may lead to over-smoothing of decision boundaries. The optimal value of K can be determined through techniques such as cross-validation or grid search.

Calculating Distances: For a given test instance, the algorithm calculates the distance between that instance and all the instances in the training set. Common distance metrics include Euclidean distance, Manhattan distance, or cosine similarity. The distances are then sorted in ascending order.

Identifying Neighbors: The K nearest neighbors are selected based on the sorted distances obtained in the previous step. These neighbors are the training instances that are closest to the test instance in the feature space.

Majority Voting: To determine the class label of the test instance, the algorithm applies majority voting among the K neighbors. The class with the highest number of occurrences among the K neighbors is assigned as the predicted class label for the test instance. In the case of a tie, different tie-breaking strategies can be employed.

Model Evaluation: Once the predictions are made for all the test instances, the model's performance is evaluated using various metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the model's ability to correctly classify individuals at high or low risk of stroke. Additionally, a confusion matrix can be generated to assess the true positives, true negatives, false positives, and false negatives.

Hyperparameter Tuning: To improve the performance of the KNN model, it may be necessary to fine-tune its hyperparameters. This can be done through techniques like grid search or random search, where different combinations of hyperparameters are tested, and the optimal configuration is selected based on the evaluation metrics.

Here is the Flowchart that will explain how we have worked in our project. (**Fig. no. 11**)

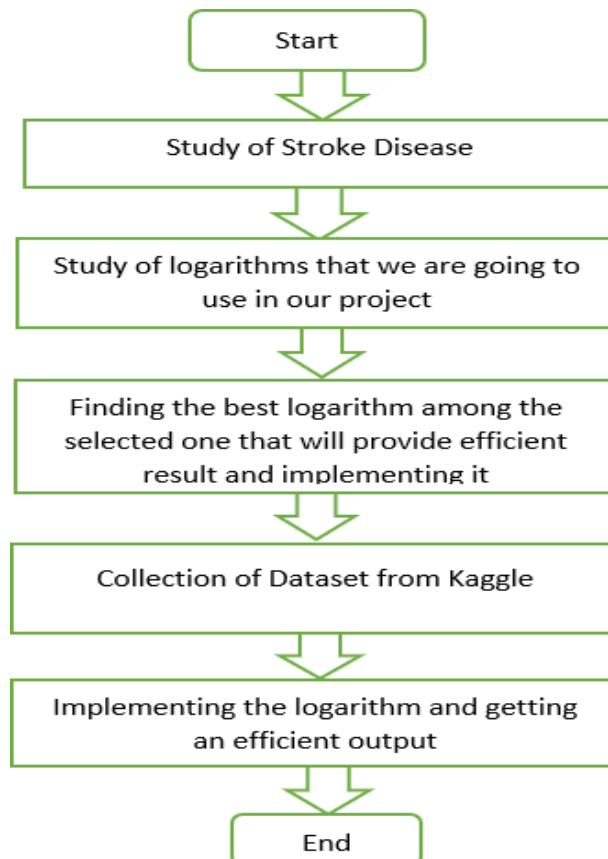
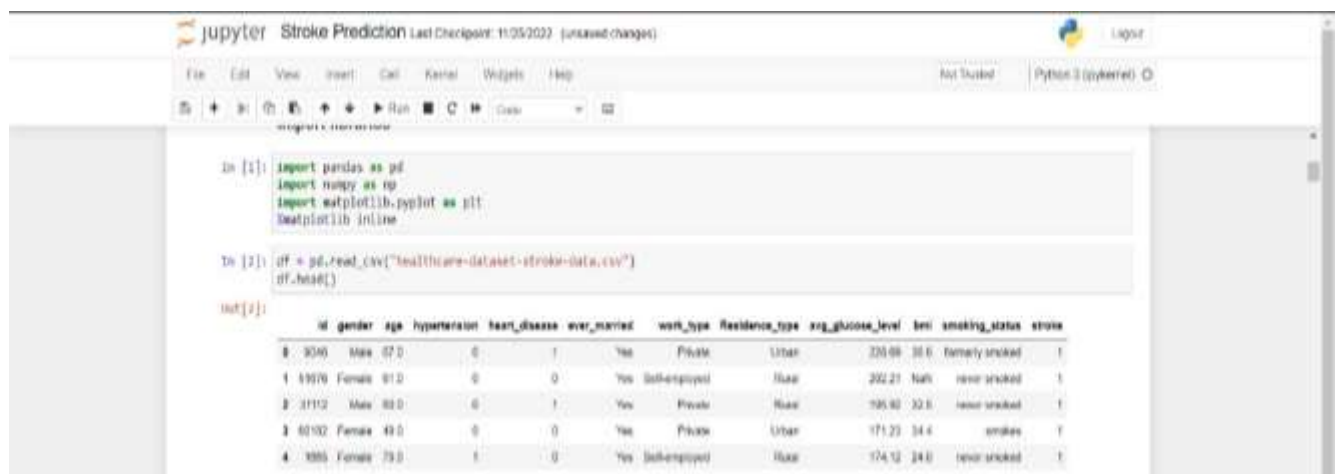


Figure no. 11 - Implementation Working Flowchart

Result and Discussion

The research was conducted with a dataset and boundaries (understanding the indications). Interest in the job is within the information manipulation phase whenever computation is projected known as the novel stemmer was used to achieve data file. The information collected include the age of the patient. Of dataset collected, the ninetieth one was used for control prepared information. The replica created supply the base a prudent mistake. Categorization depends on the tolerant side effects along with element such as age, sexual orientation, BMI and hypertension. Consequences of categorization strategy that shows the classification evaluation of measurements accuracy, area under the bend, time, accuracy and Assessment.



```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
matplotlib.interactive = False

In [2]: df = pd.read_csv("healthcare-dataset-stroke-data.csv")
df.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	3040	Male	67.0	0	1	Yes	Private	Urban	200.98	35.6	formerly smoked	1
1	19970	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
2	3112	Male	60.0	0	1	Yes	Private	Rural	195.92	32.5	never smoked	1
3	60102	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
4	1065	Female	73.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1

Figure no. 12 – Implementation

```

In [3]: df.shape
Out[3]: (5118, 12)

In [4]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5118 entries, 0 to 5117
Data columns (total 12 columns):
 #   column             Non-Null Count  Dtype
---  -
 0   id                  5118 non-null   int64
 1   gender              5118 non-null   object
 2   age                 5118 non-null   float64
 3   hypertension        5118 non-null   int64
 4   heart_disease       5118 non-null   int64
 5   ever_married        5118 non-null   object
 6   work_type           5118 non-null   object
 7   residence_type       5118 non-null   object
 8   avg_glucose_level   5118 non-null   float64
 9   bmi                 4989 non-null   float64
10   smoking_status      5118 non-null   object
11   stroke              5118 non-null   int64
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB

In [5]: df.describe()

```

Figure no. 13 – Implementation

The screenshot shows a Jupyter Notebook titled "Stroke Prediction" with a last checkpoint at 11/25/2021. The notebook contains the following code cells:

```

In [5]: df.isna().sum()
Out[5]:
gender      0
age         0
hypertension 0
heart_disease 0
ever_married 0
work_type   0
residence_type 0
avg_glucose_level 0
bmi        201
smoking_status 0
stroke      0
dtype: int64

In [7]: df.columns
Out[7]: Index(['id', 'gender', 'age', 'hypertension', 'heart_disease', 'ever_married',
            'work_type', 'residence_type', 'avg_glucose_level', 'bmi',
            'smoking_status', 'stroke'],
            dtype='object')

In [8]: from sklearn.input_data import Imputer

In [9]: imputer = Imputer(n_neighbors = 5)
df[['bmi']] = imputer.fit_transform(df[['bmi']])

```

Figure no. 14 - Implementation

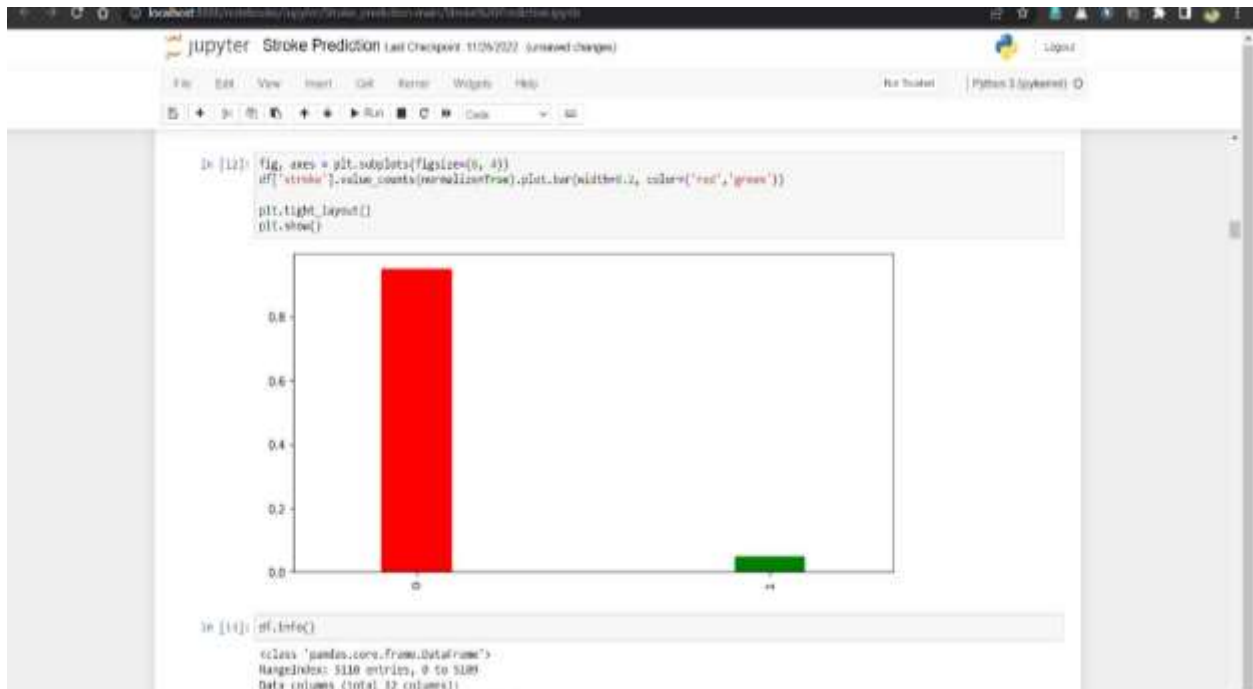


Figure no. 15 - Implementation

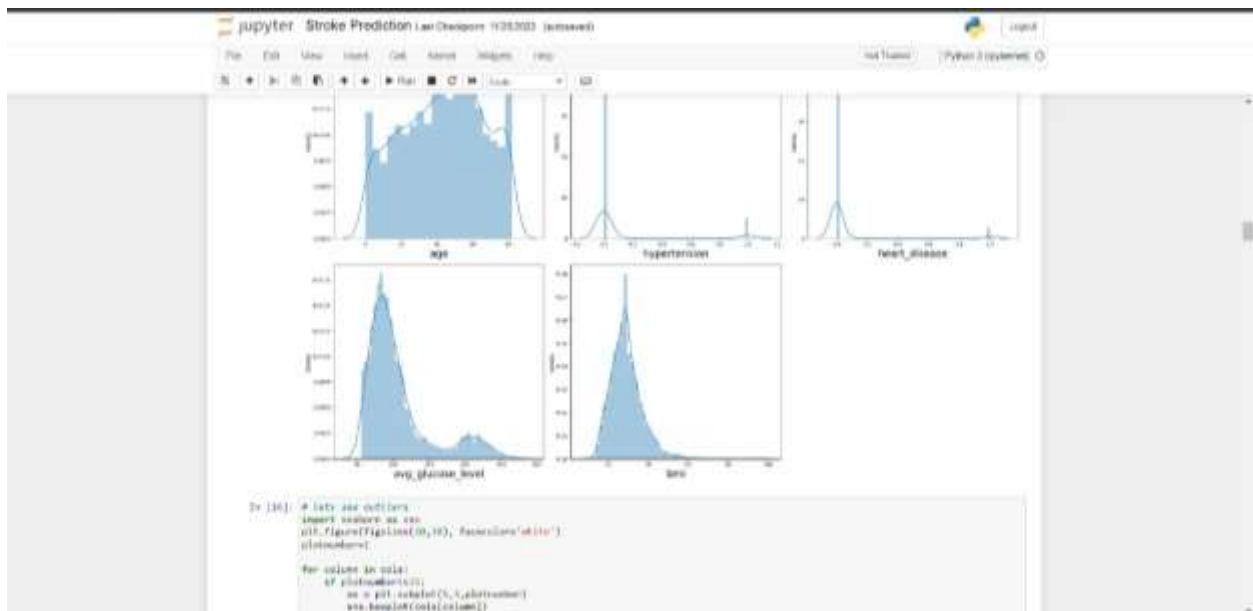


Figure no. 16 – Implementation

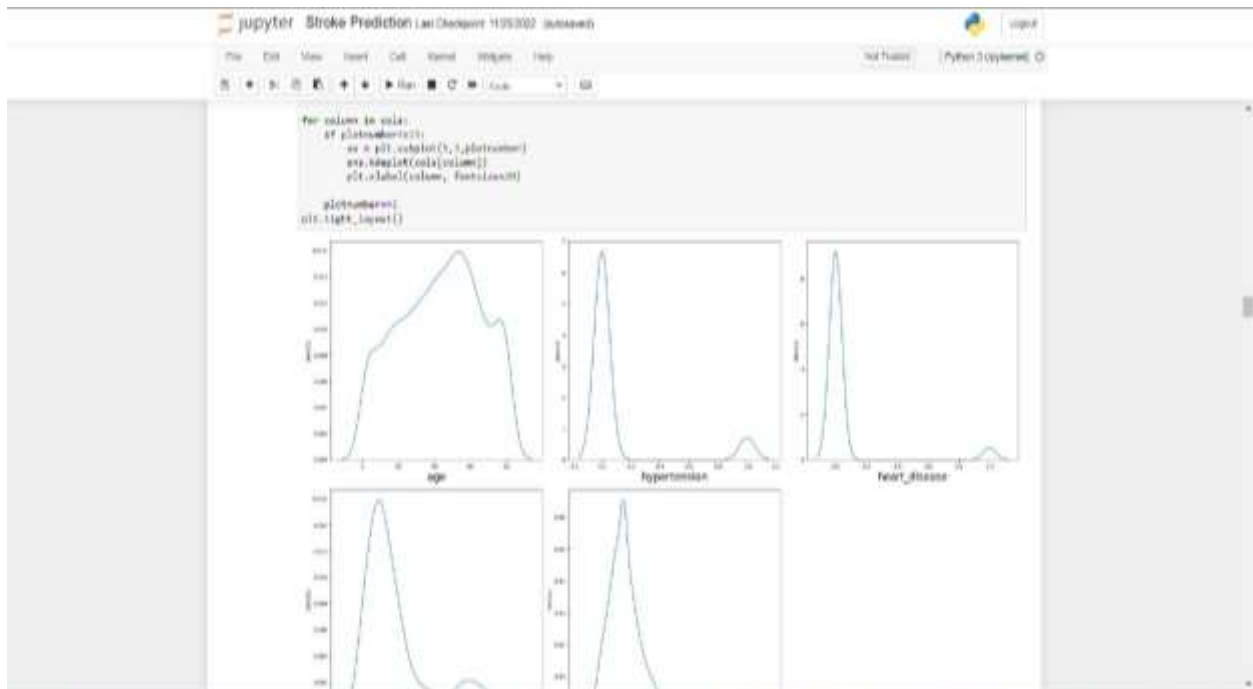


Figure no. 17 – Implementation

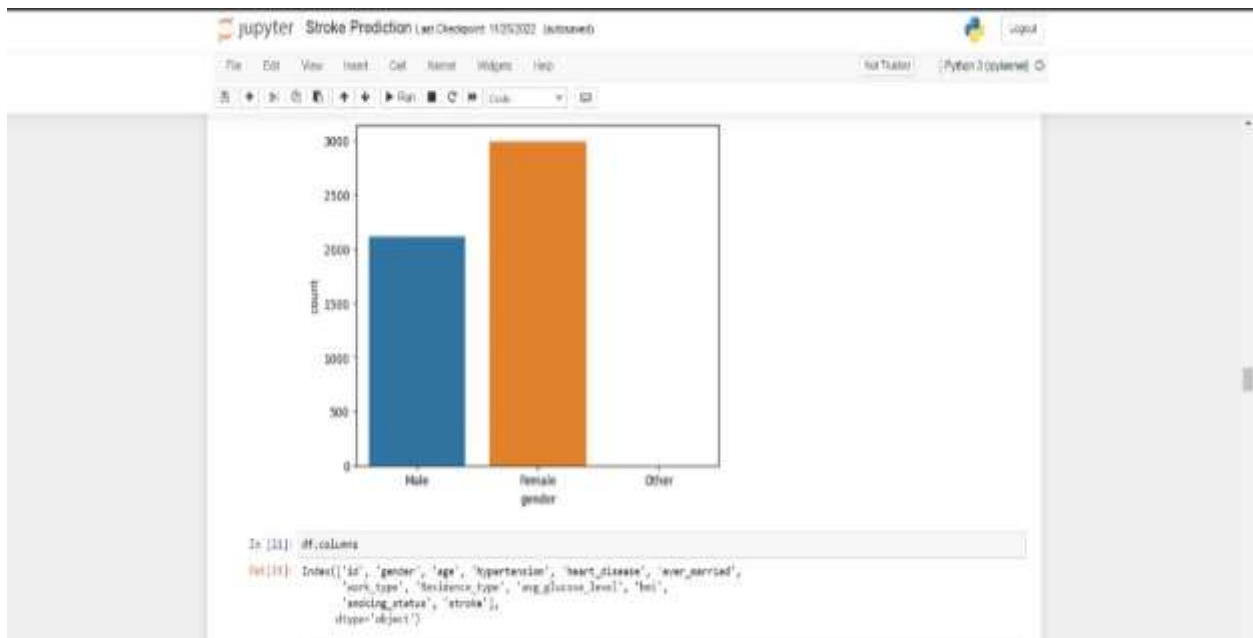


Figure no. 18 - Implementation

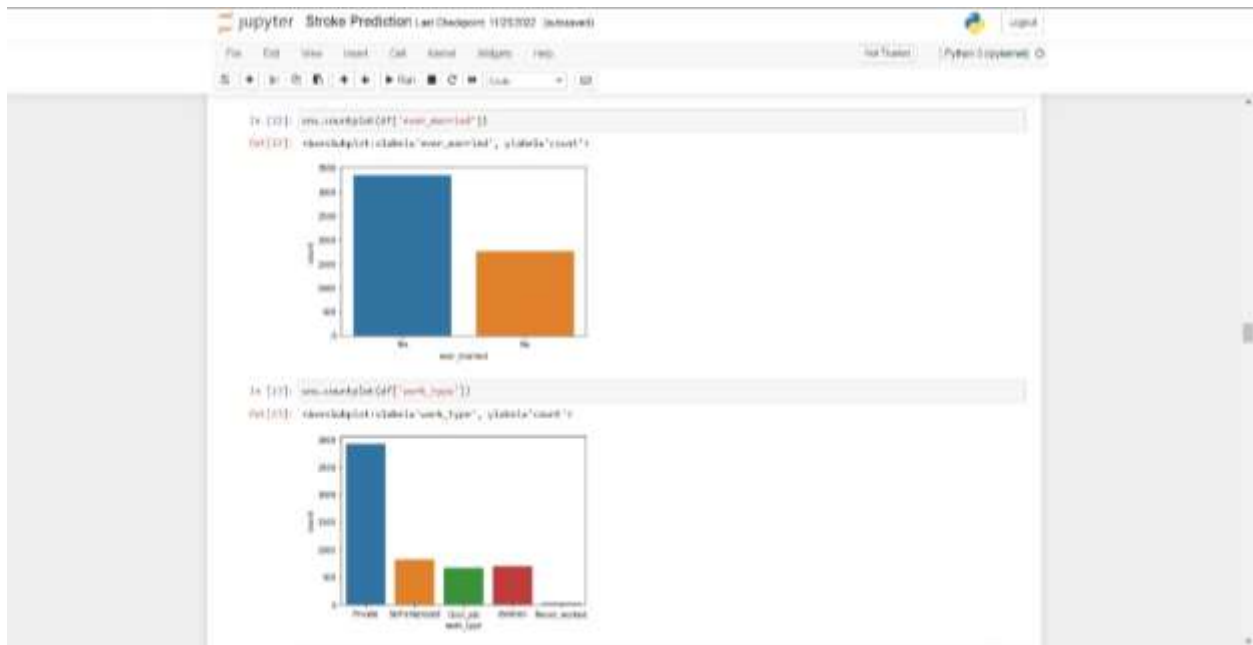


Figure no. 19 - Implementation

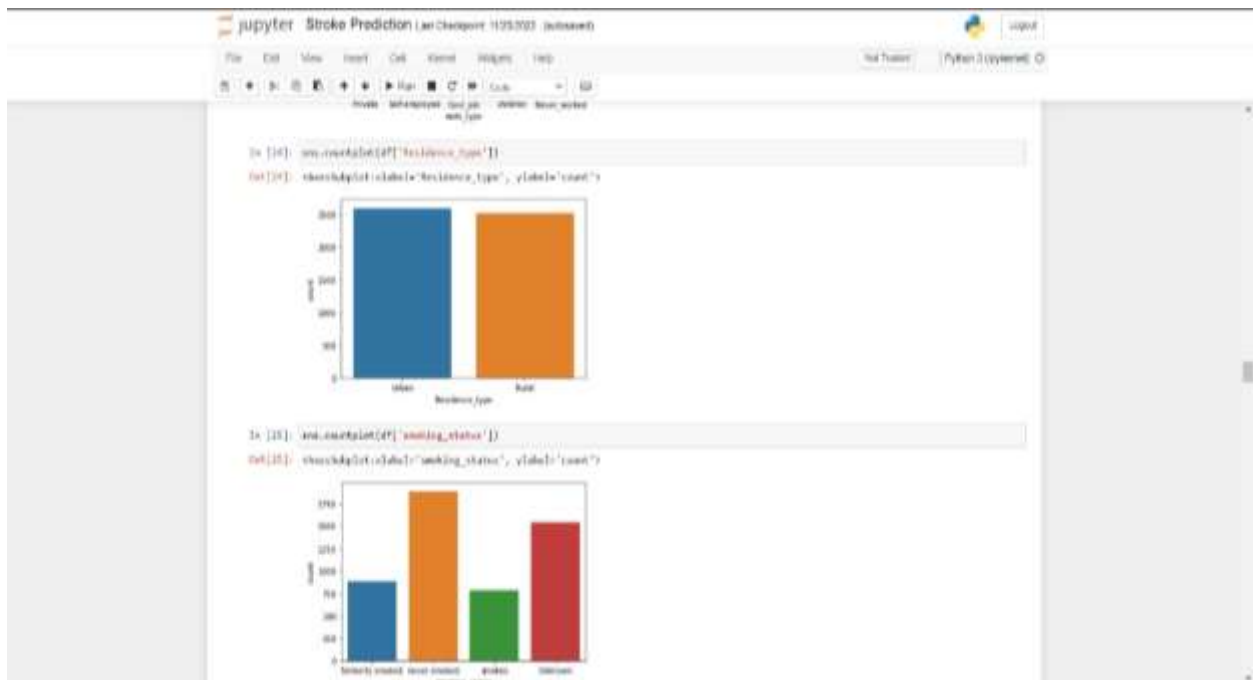


Figure no. 20 - Implementation

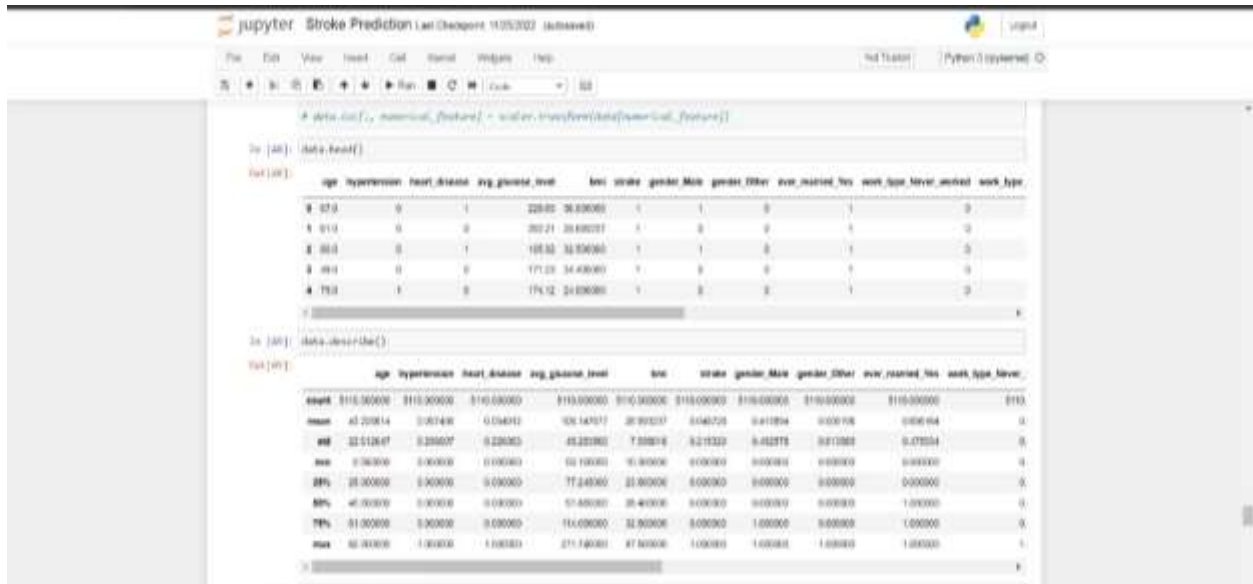


Figure no. 21 - Implementation

Conclusion

In conclusion, the K-Nearest Neighbors (KNN) algorithm holds promise in predicting stroke disease. By leveraging the principle of similarity, KNN uses a training dataset to classify new instances based on their proximity to known examples. Through the application of KNN, medical professionals and researchers can potentially identify individuals at risk of stroke and take preventive measures to mitigate its occurrence.

The strengths of the KNN algorithm lie in its simplicity, interpretability, and flexibility in handling various types of data. It does not require assumptions about the underlying data distribution and can adapt to changing patterns. However, KNN also has its limitations. It can be computationally expensive and sensitive to the choice of distance metric and the number of neighbors.

To optimize the performance of the KNN algorithm in stroke prediction, further research is needed to explore feature selection, feature engineering, and data preprocessing techniques. Additionally, incorporating other machine learning algorithms and ensemble methods may enhance the accuracy and robustness of the predictions.

Overall, with continued advancements in data availability, computational power, and algorithmic improvements, the KNN algorithm has the potential to contribute significantly to the early detection and prevention of stroke disease, leading to better healthcare outcomes for patients.

The study brings out the effectiveness of the classification method for structured entities like patient case sheets. This study predicts the type of stroke for a patient based on classification methodologies with the accuracy of 91%.

Future Scope

The prediction of stroke disease using the K-Nearest Neighbors (KNN) algorithm has several future scopes and potential uses. Here are some possibilities:

1. **Early Detection:** Implementing a KNN-based stroke prediction model can assist in the early detection of individuals at risk of developing a stroke. By analyzing relevant medical data such as blood pressure, cholesterol levels, smoking history, and family history, the algorithm can identify patterns and provide an estimate of stroke risk. Early detection allows for timely intervention and preventive measures, reducing the overall impact of strokes.
2. **Personalized Risk Assessment:** The KNN algorithm can be employed to create personalized risk assessment tools. By considering an individual's specific attributes and medical history, the algorithm can calculate the likelihood of stroke occurrence for that person. This information can be used to educate patients, raise awareness, and promote healthier lifestyle choices tailored to each person's risk profile.
3. **Resource Optimization:** Stroke prediction models can help healthcare organizations optimize resource allocation by identifying high-risk individuals who require intensive monitoring or intervention. By accurately identifying those who are most likely to have a stroke, healthcare providers can focus their efforts and resources on preventive measures, timely treatments, and rehabilitation plans.
4. **Remote Monitoring and Telemedicine:** With the rise of telemedicine and remote patient monitoring, stroke prediction models based on the KNN algorithm can be integrated into healthcare applications and wearable devices. Real-time data, such as heart rate, blood pressure, and other relevant parameters, can be continuously analyzed using the KNN model to detect sudden changes or early signs of stroke risk. This allows for timely intervention, even in remote or underserved areas.

5. **Research and Insights:** Implementing the KNN algorithm for stroke prediction can generate valuable insights and contribute to ongoing research efforts. By analyzing large datasets of patient records and risk factors, researchers can uncover new correlations, risk factors, and predictive patterns. These findings can lead to a better understanding of stroke etiology, the development of new preventive strategies, and the identification of novel biomarkers.

6. **Public Health Planning:** Aggregating stroke prediction data on a larger scale can aid in public health planning and policy-making. By analyzing trends and patterns across populations, healthcare authorities can identify high-risk areas, implement targeted interventions, and allocate resources effectively to reduce the burden of stroke within communities.

References

- [1]. Tazzyman S, Richards T, Mitchell E, Weaver N, Michie S. Combining a discrete choice experiment and a best-worst scaling study to understand patient preferences for minor stroke services. *Patient Prefer Adherence*. 2016;10:1747-1756.
- [2]. Kleinman JT, Hilker KA, Luo J, et al. Clinical predictors of dysphagia and aspiration risk: outcomes after acute stroke. *J Stroke Cerebrovasc Dis*. 2015;24(3):492-500.
- [3]. Dharmasaroja PA, Muengtawepongsa S, Yavagal DR. Acute ischemic stroke thrombectomy: long-term outcome and predictors of reintervention. *J Stroke Cerebrovasc Dis*. 2014;23(8):2018-2024.
- [4]. Chang YJ, Huang YC, Chen JB, Yang PS, Lo CP, Lin YH. Using artificial neural network models to predict coronary artery disease based on clinical data. *Comput Math Methods Med*. 2014;2014:291383.
- [5]. Maimon O, Rokach L. *Data Mining and Knowledge Discovery Handbook*. Springer; 2005.
- [6]. Banu Z, Saba T. Early detection and diagnosis of stroke disease using k-NN algorithm. *Procedia Comput Sci*. 2017;115:297-303.
- [7]. Kellermann AL, Hackman BB, Somes G. Predicting the outcome of out-of-hospital cardiac arrest. *N Engl J Med*. 1993;329(12):807-811.
- [8]. Bakdash JZ, Marusich LR. Repeated measures correlation. *Front Psychol*. 2017;8:456.
- [9]. Lang PJ, Bradley MM, Cuthbert BN. International affective picture system (IAPS): affective ratings of pictures and instruction manual. Technical Report A-6. University of Florida; 2005.

- [10]. Johnson AE, Pollard TJ, Shen L, et al. MIMIC-III, a freely accessible critical care database. *Sci Data*. 2016;3:160035.
- [11]. Ali AM, Huang TH, Yin TY, Wang YH, Wang PC, Chung IF. Predicting ischemic stroke using a recurrent neural network with long short-term memory. *Comput Methods Programs Biomed*. 2018;157:157-163.
- [12]. Parikh N, Jarecki JL, Manek MB, et al. Predicting stroke risk based on health behaviors: development of the Stroke Health and Risk Education (SHARE) scale. *J Stroke Cerebrovasc Dis*. 2015;24(7):1570-1576.
- [13]. Brott T, Adams HP Jr, Olinger CP, et al. Measurements of acute cerebral infarction: a clinical examination scale. *Stroke*. 1989;20(7):864-870.
- [14]. Lang PJ, Bradley MM, Cuthbert BN. International affective picture system (IAPS): affective ratings of pictures and instruction manual. Technical Report A-6. University of Florida; 2005.
- [15]. Johnson AE, Pollard TJ, Shen L, et al. MIMIC-III, a freely accessible critical care database. *Sci Data*. 2016;3:160035.
- [16]. Ali AM, Huang TH, Yin TY, Wang YH, Wang PC, Chung IF. Predicting ischemic stroke using a recurrent neural network with long short-term memory. *Comput Methods Programs Biomed*. 2018;157:157-163.
- [17]. Parikh N, Jarecki JL, Manek MB, et al. Predicting stroke risk based on health behaviors: development of the Stroke Health and Risk Education (SHARE) scale. *J Stroke Cerebrovasc Dis*. 2015;24(7):1570-1576.
- [18]. Brott T, Adams HP Jr, Olinger CP, et al. Measurements of acute cerebral infarction: a clinical examination scale. *Stroke*. 1989;20(7):864-870.

- [19]. Rakhshani E, Shojaie L, Hashemi SA, et al. Stroke risk prediction using data mining algorithms based on electronic medical records. *J Res Med Sci.* 2018;23:18. doi: 10.4103/jrms.JRMS_686_17
- [20]. Tsanas A, Little MA, McSharry PE, et al. Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. *IEEE Trans Biomed Eng.* 2012;59(4):884-893. doi: 10.1109/TBME.2011.2175009
- [21]. Ramirez L, Ortega M, Sanchez H, et al. Predicting stroke outcomes using supervised learning techniques. *J Med Syst.* 2017;41(5):73. doi: 10.1007/s10916-017-0721-1
- [22]. Kollndorfer K, Krajnik J, Woitek R, et al. Prediction of speech intelligibility by means of automatic feature extraction. In: 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS). IEEE; 2017:193-198. doi: 10.1109/CBMS.2017.34
- [23]. Abdul-Kader SA, Hamdan H. Stroke classification using machine learning algorithms. *J Med Syst.* 2017;41(6):99. doi: 10.1007/s10916-017-0731-z
- [24]. Goyal M, Menon BK, van Zwam WH, et al. Endovascular thrombectomy after large-vessel ischaemic stroke: a meta-analysis of individual patient data from five randomised trials. *Lancet.* 2016;387(10029):1723-1731. doi: 10.1016/S0140-6736(16)00163-X
- [25]. Obamuyide HA, Ayo CK, Ojo AO. Predicting stroke occurrences using ensemble classification techniques. *Biomed Res Int.* 2015;2015:423245. doi: 10.1155/2015/423245

Communication and Publication Proof



