RISK PREDICTION OF STROKE FROM THE RISK FACTORS USING DATA MINING TECHNIQUES

Masum Mohammed Jubayel (151-115-105)

Asima Akter Chowdhury (151-115-082)

Computer Science and Engineering

Metropolitan University

Zindabazar Sylhet

Declaration

We hereby declare that the thesis is our original work and it has been written by us in its entirety. We have duly acknowledged all the sources of information which have been used in the thesis.

The thesis has also not been submitted for any degree in any university previously. Our paper was accepted and successfully presented in 6th International Conference on Natural Science and Technology (ICNST'19) Chattogram, Bangladesh. The accepted and presented papers submitted to *iLab Australia* series as well as other Abstracting and Indexing (A&I) databases for indexing.

-------------------------------- ----------------------------------

Asima Akter Chowdhury Masum Mohammad Jubayel

Recommendation Letter from Thesis Supervisor

These students, Asima Akter Chowdhury and Masum Mohammed Jubayel whose thesis entitled “Risk Prediction of Stroke from the risk factors Using Data Mining Techniques”, is under my supervision and agrees to submit for examination.

Rahatara Ferdousi

Lecturer,Department of CSE,

Metropolitan University.

Date: ______________________

…………………………….

Signature

Qualification Form of Bachelor Degree

Students Name: Asima Akter Chowdhury, Masum Mohammed Jubayel

Thesis Title: “Risk Prediction of Stroke from the risk factors Using Data Mining Techniques”

This is to certify that the thesis submitted by the student named above in April, 2019. It is qualified and approved by the Thesis Examination Committee.

----------------------------	------------------------------------	----------------------
Head of the Dept.	Chairman, Thesis Committee	Supervisor

Acknowledgment

We are thankful to our supervisor “Rahatara Ferdousi” Lecturer, Department of CSE, Metropolitan University. For her support and direction to our working progress. We also thanks Sylhet Woman Medical Collage and Hospital and Jalalabad Ragib Rabeya Medical College and hospital from where we collected our data for our thesis work and other websites for helping us to complete our thesis.

Abstract

Stroke, a fatal non-communicable disease of any age, kills more people than AIDS, Tuberculosis and Malaria put together in each year. WHO estimated around 6.2 million deaths because of stroke in 2008. As the incidence, prevalence, mortality, and disability rates are increasing, overall stroke burden has increased globally. Almost 70% of patients are unaware of their mild stroke, 30% seek medical attention lately and another 30% suffer from recurrent stroke, before seeking attention. Data mining, with its several techniques for classification and regression, plays a leading role in developing an effective model of risk prediction in the context of healthcare. Even though stroke prevention is a complex medical issue, primary prevention could be feasible by using data mining classification techniques that will assess risk factors to predict the likelihood of the disease among mass people. This work is aimed at providing an analysis of different data mining classification algorithms like Naïve Bayes (NB), Decision Tree (DT), Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Classification and Regression tree (CART)etc. on a newly created dataset of 435 patient's risk factors to find the algorithm with the best accuracy to propose a tool for the end users to check stroke risk prediction.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

1.1 Introduction

Stroke, a disruption of blood supply to the brain due to blockage (blood clot) or rupture of blood vessels that causes the death of some brain cells as a result of lack of oxygen supply to them[1][A] Stroke is one of the leading causes of death, physical disability, dementia, and depression around the world. The incidence of stroke increases with age and the presence of atherosclerosis along with smoking tobacco, obesity, physical inactivity, hypertension, alcohol intake, diabetes, high blood lipid level, kidney disease, unhealthy diet, stress, male gender, genetic factor etc.[B] The overall incidence of stroke had exceeded 20% in low to middle-income countries than that of high-income countries from the year 2000 to 2008. It is estimated that globally 1 in 6 people will be affected by stroke in their lifetime. [b] When it happens, the brain cells divested of oxygen and brain cells begin to die. Then the abilities controlled by that area of the brain such as memory and muscle control are lost. Some strokes affect the muscles used to urinate. There are two types of stroke. Those are Ischemic Stroke and Hemorrhagic Stroke. An Ischemic stroke is the most common type. It occurs when blood supply is cut off to part of the brain. It accounts for the majority of all strokes. Approximately 85% of strokes are ischemic caused by vascular occlusion. An ischemic can occur because of lesions caused by atherosclerosis. These lesions may form in the small arteries of the brain and they can block blood flow to the brain. A hemorrhagic stroke is a different kind of stroke caused by bleeding in the brain. It happens when a blood vessel breaks and bleeds into the brain. Blood spills into or around the brain and creates swelling and pressure, damages cells and tissue in the brain. Hemorrhagic stroke is caused by a rupture in a weakened blood vessel in the brain. Hemorrhagic stroke account for about 20% of all strokes. The brain is one of the largest and most complex organs in the human body. It is made up of more than 100 billion nerves that communicate in trillions of connections called synapses. [1] It is the central organ of the human nervous system. It controls most of the activities of the body, processing, integrating and coordinating the information it receives from the sense organs and making decisions as to the instructions sent to the rest of the body. Because of stroke, brain cells start to die or damage. So, symptoms occur in the body parts (face, eyes, arms, legs etc.) that these brain cells control. There are number of factors which increase the risk of developing stroke such as age, gender, high blood pressure, diabetes, ischemic heart diseases, smoking, family history of stroke, stress and depression, overweight, obesity, abnormal cholesterol levels and so on.

About 70% of patients suffering from stroke fail to recognize the initial symptoms and 30% seek medical attention lately regardless of their demographic groups. An interactive and easily accessible end-user tool will be very effective to make the people acquainted with stroke-related risk factors and common sign symptoms for acute prevention and minimize further long- term physical problems as well as financial burden. [C]

In this thesis paper, we are aimed at providing a model for the risk prediction of Stroke. At present, data mining techniques will help us a lot to predict risk. Data mining techniques which include classifications, clustering, association rule mining for finding risk prediction. In this research work, the Naïve Bayes, Support Vector Machine (SVM) classifier algorithm is also used for stroke risk prediction.

1.2 Motivation

Bangladesh is a developed country. In our country people are not more conscious for their health. Many of our people do not follow healthy diet. Therefore, they increase their risk of developing stroke. There are many people who have the problem of high blood pressure which is also responsible for developing stroke. So, in our thesis paper we wanted to predict the risk of developing stroke in the early stage. If a person can know that he/she is in the risk of developing stroke then she/he will be more careful than the past. It will be sound good for our people. Data mining classification technique has become a well accepted approach for risk prediction. So, we chose it and our supervisor also encourage us to do this thesis work.

1.3 Objectives and Contributions

We are living in the modern era. In our present time, the use of mobile phones is huge. We wanted to propose a reliable tool that can be a mobile app or a website. With the help of this tool, a user can predict his risk level of developing stroke in the early stage. For this we need to find out one best classification algorithm for prediction. This tool will help the user to predict risk of stroke, as well as provide some healthy tips.

- To prepare an appropriate dataset of risk factors.
- To use the most popular risk prediction data mining algorithm for the dataset.
- To obtain the best algorithm.
- To design an end-user tool (website) for mass people with the risk feature developed with the best algorithm.

1.3 Thesis Organization

The remainder of the thesis is organized as the following:

Chapter 1 (Introduction): Presents an overview of the disease diabetes and its impact on the current world. Afterwards, we explained the reason that motivated us to do the thesis. We also explained our objectives and contributions.

Chapter 2 (Background Study): Presents an overview of background knowledge and technical aspects. Background concepts of data mining (DM) and knowledge discovery of database (KDD).

Also, an overview of all the supervised and unsupervised learning algorithms and how they are implemented in real-world problems.

Chapter 3 (Related Works): Presents several existing disease detection techniques, how they are implemented by different authors and their working process along with acquired results from their experiment.

Chapter 4 (Methodology): Proposed our idea and elaborates on the system design phase. The design phase includes the proposed architecture techniques, feature selection, tools and selected algorithms.

Chapter 5 (Experimental Analysis): This section contains description and details of the dataset along with the results that we got from training and testing the dataset. It also contains a proposed method (websites) that can be used by a person to detect the potentiality of he/she having stroke.

Chapter 6 (Conclusion): Finally, the thesis is concluded in this chapter with suggestions for future research.

CHAPTER 2

Background Study

According to the World Health Organization, 15 million people suffer stroke worldwide each year. Of these, 5 million die and another 5 million are permanently disabled. High blood pressure contributes to more than 12.7 million strokes worldwide. Europe averages approximately 650,000 stroke deaths each year. In developed countries, the incidence of stroke is declining, largely due to efforts to lower blood pressure and reduce smoking. However, the overall rate of stroke remains high due to the aging of the population. [2]

2.1 Data Mining

Data mining is a process of collecting or gathering hidden data from large data sets where data set means a collection of data. Data mining process is used for finding a pattern or similarity and relationship among large data set. We are living in an age where data is constantly being collected. Our required and unnecessary data is being submitted too. So, we need to get help with data mining if we need to find our required data. In data mining process, Flat Files, Rational Databases, Datawarehouse, Transactional Databases, Multimedia Databases, Spatial Databases, Time Series Databases and World Wide Web (WWW) are different sources are used. Classifications, regression, clustering analysis, associations are different functionalities of data mining. Data mining techniques have great success in search engine, health care, education and business intelligence etc. It is also a very helpful techniques for researcher. So, we wanted to use it in healthcare sector to predict the risk of stroke.

Data miming techniques have a great success in research and development. Data mining process can be thought as a genuine appraisement of information technologies. There are many different issues in research which can be implemented by using data mining process and techniques. [3] In health care sector, a huge amount of data is adding day by day, those data can be mined for different research work. So, we preferred to use data mining techniques in our thesis work.

Knowledge Discovery in Databases (KDD)

As we know in medical sectors there are many data sets are being collected and we do not need all of those data sets. So, we have to find out our necessary data sets among them. Knowledge Discovery Database can help us in that situation. Knowledge Discovery Database is a name of a process that can recognized our required data from the large data sets. It is a very helpful process for finding a pattern form large amount of data.

Data mining is a part of KDD. In KDD process at first data that we have collected through database from the user interface, will be stored in data warehouse. In warehouse data will be checked or tested that whether this data is good or not for the user or the operation. Then data will go through many different phases namely; data mining, pattern finding, related data or pattern finding, representation of data. KDD starts it's work through data cleaning, data integration, data selection. **Data Cleaning** is a task preformed in KDD for removing noise, handling missing data, finding error, inconsistent data etc. For example, data parsing in which data will be checked weather the data is acceptable or not.

Data Integration is a task used for integrating or collecting together all of those data which are coming from different sources for storing that's data set in data warehouse.

Data Selection - KDD will retrieved all of the data which are related to our required data sets from the database for our analyzation.

After this KDD complete the rest four steps. They are transformation, mining, interpretation or evaluation and knowledge.

Transformation is a step of KDD where all data which are cleaned, integrated and selected will be being consolidated or transformed from one format to another format. With the help of this step the result of data mining in KDD will be more accurate and data will be easier to understand. For transforming data into a form KDD follows some steps those are smoothing, attribute constructing, aggregation, normalization, discretization, hierarchy generation for nominal data etc. It is a very important step.

Data Mining is a most important step in the whole process. In there, data will be mined to find a pattern or similarity. After finding the pattern, it will be evaluated and interpreted in **Data interpretation or evaluation** steps for making the pattern more appropriate. Then KDD will show or represent the data sets to the user which is known as knowledge representation it can be as like image, graph, tree, chart, text etc. If the represented knowledge is matched with the user required data then KDD will store it as **knowledge**.

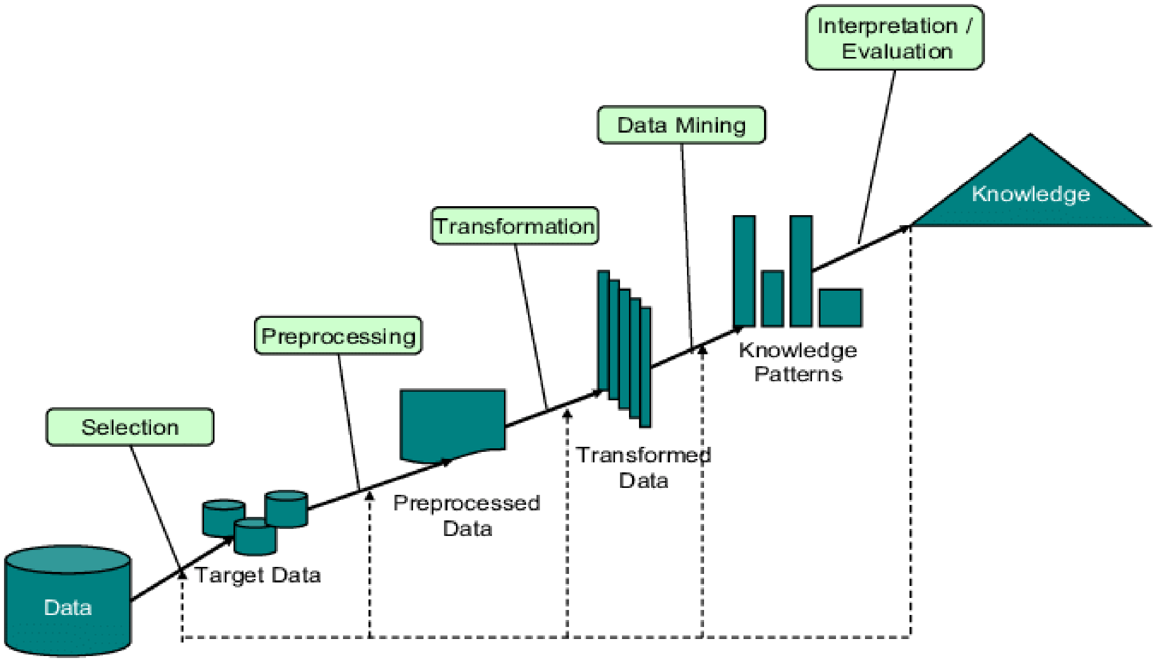


Figure 1: Steps of KDD

2.2 Classification

Generally, classification means classify something based on its some property or its identity. In machine learning algorithm, classification use to predict the label or the class by using a classification model. Supervised learning is the origin of the classification. Classification includes two parts. First one is learning part. A classification model is consecrated by using the training data set with the help of some classifier rules in this part. The second one is classification part in where the established model is used to predict the class (label) on the test data. Before using the constructed classification model to predict the class we need to find out the accuracy of this model. For finding accuracy at first, the mode

There are a lot of algorithms are available in classification. We can choose algorithms from there depending on the nature of our data

As classification process is used to predict the class label so the researcher used it in many times to predict the risk factors of many diseases in different research works. In machine learning, we have two types of learning methods those are supervised and unsupervised learning. Classification goes to supervised learning and clustering goes to unsupervised learning. The machine learning can give better performance with the help of both supervised and unsupervised learning. To find the patterns from the given data sets and to give the best decision by observing those data sets the process works. Now we are describing about some different algorithms used in both learning methods in prediction.

The two learning methods are-

1) Supervised learning
2) Unsupervised learning

2.3 Supervised Learning

Supervised learning is a learning method where the training data set is available to built a classifier model to predict the output. It is used to predict the future output where the past data set are well defined. If anyone wants to determine the class the label from a scenario which is optimal for an unseen instance then the supervised learning methods also can do this correctly. In supervised learning

-known class available.

- the both input and output data are also known to us so we can check whether the given output is matching to our target output or not.

Supervised learning has great success in many different fields such as bioinformatics, cheminformatics, database marketing, handwriting recognition, spam detection, pattern recognition, optical character recognition etc.

In supervised learning many different algorithms are available. Those are Support Vector Machines, Linear regression, Logistic regression, Naïve Bayes, Decision trees, K-nearest neighbor, Neural Network and so on.

2.3.1 Support Vector Machine

Support Vector Machine is a training algorithm works for pattern recognition problem which is a supervised algorithm also. It can work for both linear and non-linear data by finding a hyperplane for separating the data sets into two classes in the input space. Generally, SVM is a non-probabilistic binary classifier method use for finding the result in the simplest way which was based on a strong mathematical foundation. At first, it works for finding the hyperplane so that it can separate the given data set into the class labels those data belongs. For finding that hyperplane, the SVM creates some margins and checks which one is the best margin between all.

For choosing the best hyperplane, the machine checks which line is passing as far as possible from all input data. Because if the machine chose the line which one is very close to the data then the line will not be very perfect for generalizing correctly.

SVM works in two steps. First step is training and the second step is classification step. If the data is linear, in training step SVM selects the support vector for both positive and negative class and then execute the training algorithm. For constructing the hyperplane, it uses linear discriminant function, y = (w*x) + b.

Where,
y = are labels which class the input data belongs
x = are given input data instance
w = is the weight vector
b = is the offset.

For non-linear data SVM uses kernel function and mapped the input data into high dimensional space so that the input non-linear data set becomes linear and then the SVM works. It uses some kernel function for non-linear data to transform into higher dimension space those are Polynomial kernel, Gaussian radial basis function kernel, Sigmoid kernel etc. It is very faster and powerful binary classifier method. It can generalize pattern recognize problem well even when the size of given data set is big. It has nice math property and also have the ability to handle large feature space and we can use different types of data as input like trees, strings, pixel maps etc. into SVM. Although SVM is a good supervised training approach, it can consider or give the output only for two classes. It also needs a good kernel function for non-linear data weather it cannot give the best result. Dr. S. Vijayarani and Mr. S. Dhayanand showed in their research paper which was published in 2015 that the accuracy of SVM is better than Naïve Bayes algorithms [6].

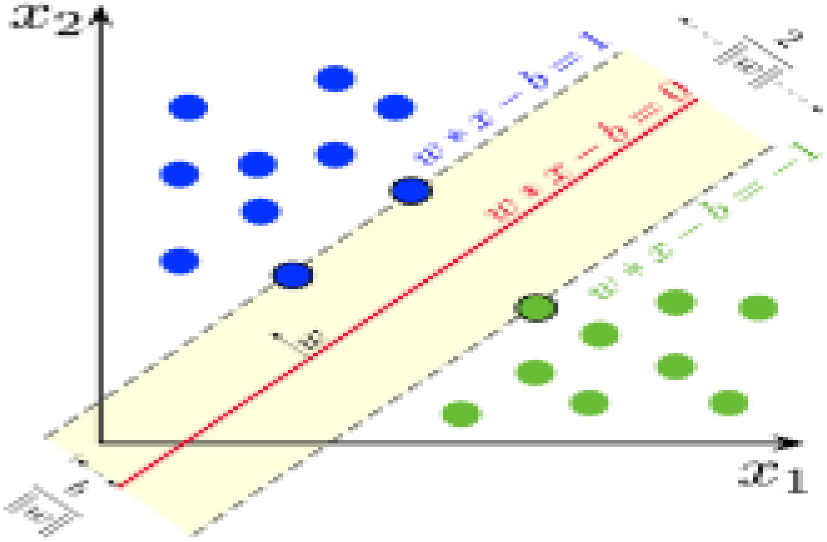


Figure 2: Support Vector Machine (SVM)

2.3.2 Naïve Bayes

Naïve Bayes is a type of classification which follows Bayes theorem and also known as probabilistic classifier method. In 18th century Thomas Bayes who was a nonconformist English clergyman worked in probability and decision theory [5]. Naïve Bayes algorithm use to predict the probability of the result or the output for an unseen or unlabeled test input. As it is a type of supervised learning method so, the training data set will available for this algorithm. It is very helpful for large data set and very easy to implement. It works with three step those are prior probability, conditional probability and posterior probability.

First step is **prior probability**. In this step the data set will convert into a frequency table. Second step is **conditional probability**. In this step, a table will generate for each train data depending on the decision class attribute. Third step is **posterior probability** and here if decision class attribute has two labels than prior will count for both class and check the value of which class is greater and finally predict the decision. Suppose we have three attribute class and one decision class. The decision class has two labels those are positive and negative and we also have a test data set which is unseen in the train data set. Naïve Bayes algorithm uses to predict the probability of the result or the output for an unseen or unlabeled test input. There are three equations are given below which represents the classification formula of Naïve Bayes classifier. Here pos and neg represent respectively with the risk of having stroke and without the risk of having stroke.

P(pos) = Total positive risk factors.

P(neg)= Total negative risk factors.

test dataset, $T_{test} = a_1, a_2, a_3 \ldots\ldots\ldots\ldots\ldots\ldots a_n.$

$$P\ (Pos \mid T_{test}) = P\ (a_1|pos) * P\ (a_2|pos) * P\ (a_3|pos) * \ldots\ldots\ldots\ P\ (a_n\ |pos) *P\ (pos) \qquad (1)$$

$$P\ (Neg| \ T_{test}) = P\ (a_1|neg) * P\ (a_2|neg) * P\ (a_3|neg) * \ldots\ldots\ldots\ P\ (a_n\ |neg) *P\ (neg) \qquad (2)$$

$$P\ (\text{Ttesti}\ |pos) = \frac{Total\ (\text{Pos} \mid \text{Ttesti})}{\text{Rn}} \qquad (3)$$

Where n is the total number of attributes and i is an incremental number it reaches until the last number of attributes of test dataset and Rn represents the total number of risk factors.

Naïve Bayes classifier method has great success in text categorization, hand writing recognition, medical diagnosis, spam filtering, able to handle continuous data, good for both binary and multi-class classification tasks etc. Aiswarya Iyer, S. Jeyalatha et.al. made an experiment in their research paper named “Diagnosis of diabetes using classification mining techniques” and showed that the Naïve Bayes algorithm gives the better performance accuracy than J48 decision tree [8].

2.3.3 Decision Tree

A decision tree is like a flowchart or a graph includes with a root node, leaf nodes. It uses internal node for representing attributes or features, a decision rule is represented by its branch and for representing the outcome or result it uses leaf node. It is a decision support system tree that is one of the most popular and frequently used supervised learning algorithm. This supervised learning algorithm can work good for both classification and regression tasks. Although it can have more than one leaf nodes and decision nodes, it can have only one root node. It is drawn upside down and its root node is at the top. Decision tree selects the most important attribute from the given dataset as its root node. For making better decision, predicting the possible outcome, decision tree is a very easy and widely accepted learning algorithm. It can perform very well in KDD and data mining by enabling the model and knowledge extraction from the given dataset. It is able to perform with missing data or value. It can handle many types of input dataset like Numeric, Textual and Nominal. Within small number of efforts, it can give high performance also and it is very easy to understand not only for the developer or researcher but also for the end user.

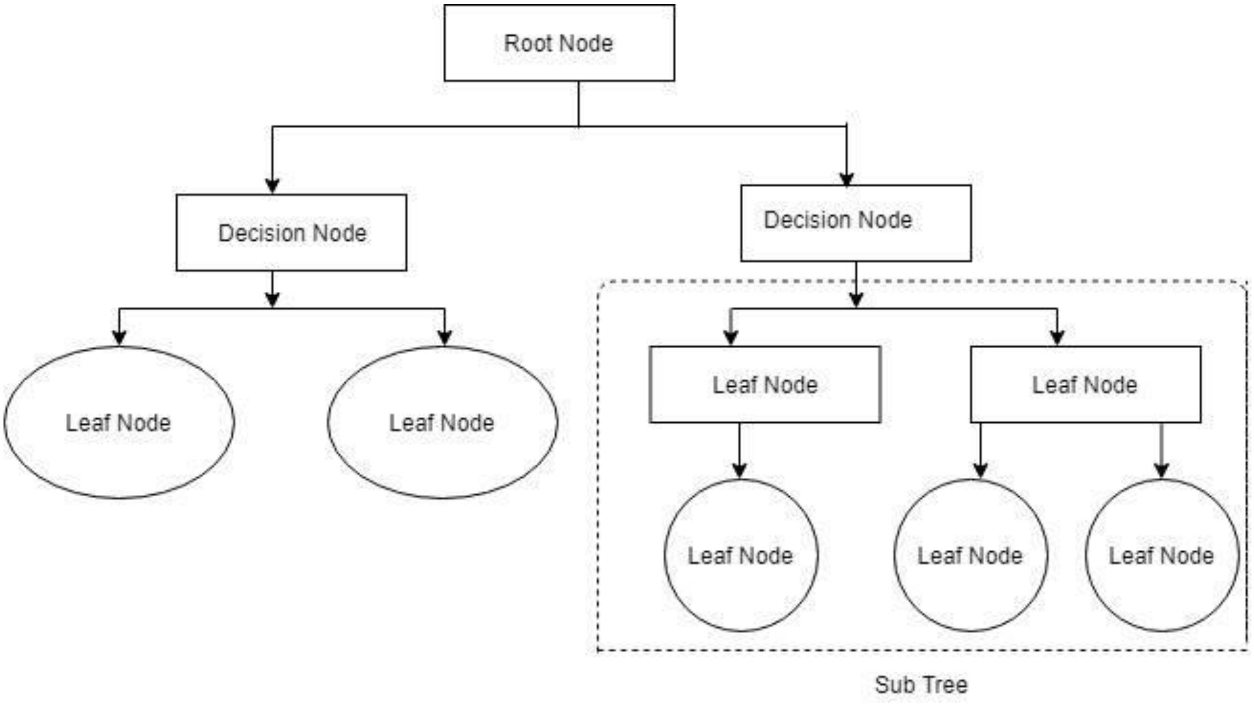


Figure 3: Decision Tree

Decision tree algorithms represents itself by following tree structure. By using ASM (Attribute Selection Measure) at first it selects the best attribute to split the records from train dataset. It uses the selected attribute for making a decision node and then breaks the dataset into smaller subsets.

It repeats the process for each and every child recursively until all the tuples belongs to the same attributes value or there are no more instances or attributes are remaining. Then it builds a tree or model. For test dataset, it evaluates the provided the model and measures the performance evaluation by using the formula of accuracy, precision and recall. The figure 4 shows how the decision tree algorithm works.

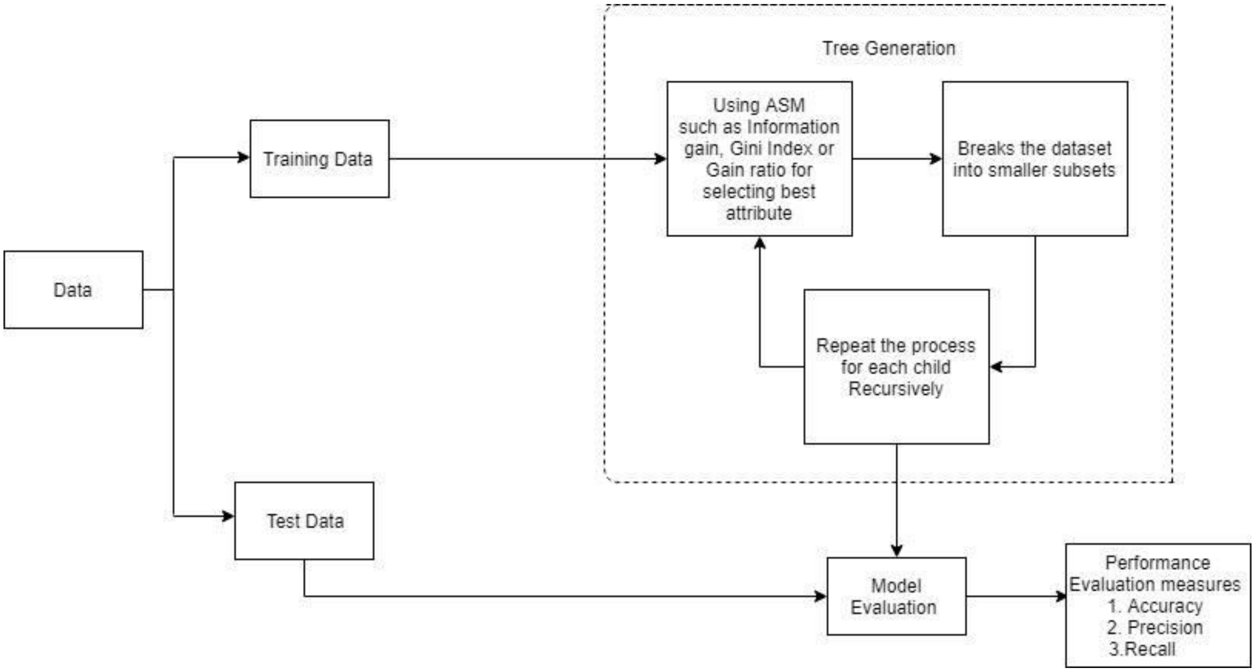

Tree Generation
Data
Training Data
Test Data
Using ASM such as Information gain, Gini Index or Gain ratio for selecting best attribute
Breaks the dataset into smaller subsets
Repeat the process for each child Recursively
Model Evaluation
Performance Evaluation measures
1. Accuracy
2. Precision
3.Recall

Figure 4: The working process of decision tree algorithm.

For selecting the best attribute, it uses ASM (Attribute Selection Measures) and for continuous-valued it uses some most popular attribute selection measures like Information gain, Gain Ration and Gini index.

Information Gain is one of the most popular attribute selection measure which measures the entropy also. Entropy prescribed as the impurity in a group of examples in the information theory. Based on the given dataset, information gain computes the difference between the entropy before split and the average entropy after split. Information gain is used by ID3 decision tree. There are three equations from (4) to (6) are given below. Information gain follows those equation for measures.

$$\text{Info(S)} = -\sum_{i=1}^{k} \text{pi} \log_2 \text{pi} \qquad (4)$$

Where, Pi is the probability that an attribute tuple S belongs to class Ci.

$$\text{Info}_{\text{A}}\text{(S)} = \sum_{j=1}^{V} \left(\frac{|\text{Sj}|}{|\text{S}|} * \text{Info(Sj)} \right) \qquad (5)$$

$$\text{Gain (A)} = \text{Info(S)} - \text{Info}_{A}(S) \qquad (6)$$

Where, Info(S) is the required information for identifying the class label within S attribute tuple, Info $_{A}$(S) is expected information, $\frac{|Sj|}{|S|}$ is the weight of the jth partition and i is an incremental number .Here, attribute A will be selected within the highest range of value of Gain (A) as the splitting attribute.

2.3.4 J48 Decision Tree

J48 is one of the most easy and simple algorithms in terms of implementation. It is a most preferable algorithms in supervised learning. At first, it broken down it's train dataset into smaller subset by using decision tree and at the same time it creates a decision tree which is incrementally developed. J48 decision algorithm is available in WEKA data mining tool. J48 decision tree algorithm is an implementation of java implementation of the C4.5 algorithm. It uses gain ratio as an extension of information gain. By normalizing the information gain using split info, gain ratio can handle the issue of bias. If v is the number of discrete values here then split info and gain ratio can be define by following equation (5) and (6) which are given below:

$$\text{SplitInfo}_{A}(S) = -\sum_{j=1}^{v} \frac{|\,Sj|}{|\,S|} * \log_{2} \frac{|\,Sj|}{|\,S|} \qquad (7)$$

$$\text{GainRatio (A)} = \frac{Gain\ (A)}{\text{SplitInfoA}(S)} \qquad (8)$$

Where, $\frac{|\,Sj|}{|\,S|}$ is weight of the jth partition and the best attribute will be selected with the highest gain ratio.

2.3.5 Classification and regression tree

Leo Breiman introduces a decision tree algorithm named classification and regression tree in short CART for solving the predictive modeling problems [38]. This decision tree algorithm provides a foundation for many others important decision tree algorithms like random forest, bagged decision trees, boosted decision trees. In CART algorithm, leaf node represents the output and that is used for predicting where root node represents the input of the dataset. For making prediction with the help of binary CART algorithm is relatively straightforward. If a person wants to create a CART model from the given dataset then he will divide the input space, select the input variable and split point on those variables. This process needs to be done until a suitable tree is being constructed. Recursive binary splitting approach is used for diving the space. CART model used the Gini Index method for creating the split points. It follows the equations (9) to (11) for splitting which are given below:

$$\text{Gini (S)} = 1\text{-}\sum_{i=1}^{m} Pi^2 \qquad (9)$$

Where, S is the sample, Pi is the probability that a tuple in S belongs to class Ci.

If a binary split on attribute A partitions the sample or data S into S1 and S2. The Gini Index of S is:

$$\text{Gini}_A\text{(S)} = \frac{|\,S1|}{|\,S|}\text{Gini(S1)} + \frac{|\,S2|}{|\,S|}\text{Gini(S2)} \qquad (10)$$

For selecting the splitting attribute, we need to find the subset which give the minimum value or smaller value of the Gini index. So, we need to follow the equation (11) for finding the attribute with minimum Gini index.

$$\Delta\ \text{Gini (A)= Gini (S)- Gini}_A\text{(S)} \qquad (11)$$

Where, A is the attribute with the minimum value of Gini Index is chosen for splitting attribute.

2.3.6 Random Forest

The random forest consists of many trees, and it makes a prediction by averaging the predictions of each component tree. In every iteration of the algorithm it constructs a random decision tree and after a large number of decisions tree is created, the outputs are aggregated and the result represents a strong ensemble.

The Random forest has three key advantage – firstly random forest can be used for both classification and regression, secondly, over-fitting make the result worse, but if there are enough trees in the forest, the classifier model can avoid over-fitting. Thirdly, Random Forest can handle missing values, and it can be modeled for categorical values.

2.3.7 Logistic Regression

It is one of the simplest techniques for constructing classifiers. Naive Bayes classifiers are a family of simple probabilistic classifiers which uses Bayes theorem with strong (naive) independence assumptions [9]. In simple word, Naive Bayes classifier finds the probability of being of an object in clusters based on given train dataset and the class that has high probability. It adds this object to that cluster. For example, if we give a train data set which says apple is red and mango is green then whenever it will find apple it will put that in the red color group.

2.3.8 Artificial Neural Network (ANN)

It is a system inspired from the brain, constructed for the purpose to replicate the way human learns. It consists of input and output layer and a hidden layer that is shown in Fig. 5, which consist of units that transform input layer into something that the output layer can use [41]. ANN works very well in finding patterns that are very complex or difficult for a programmer to extract and teach the machine to understand and recognize. A diagram below shows the layers of ANN-

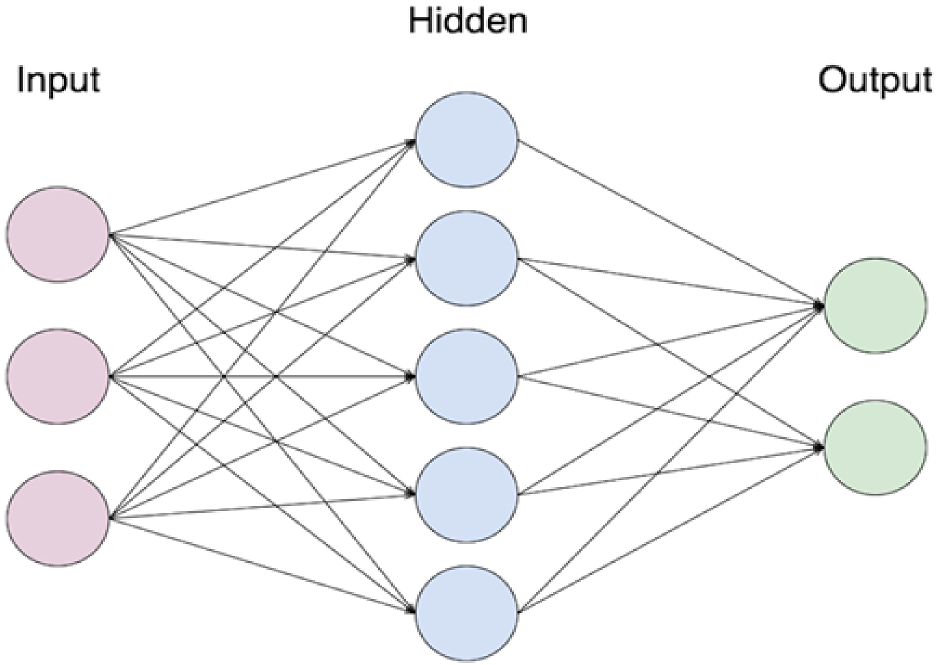


Figure 5: Multiple Layers of Artificial Neural Network

2.3.9 K-Nearest Neighbors

K nearest neighbors is a non-parametric technique lazy learning algorithm used in various data analysis implementation such as pattern recognition, database, mining of data due to its peak accuracy and simplicity. Non-paramedic means the structure of the model decided from the dataset and it is called lazy learning because it does not need any training data for model creation, it is done in the testing phase. It is counted under top 10 algorithms in data mining [39].

In KNN the letter ‘K’ represents its nearest neighbors. The main deterministic factor is of KNN is its neighbors. To predict a label of an unknown point P its closest neighbors K points need to find and classify the point P by most votes given by its neighbors [40]. Every individual object vote for their class and the class with the most votes are taken as a prediction. Euclidean distance, Hamming distance, Manhattan distance are used to find the distance between points e.g. A and B in which the Euclidean distance function is the most widely used one. For points A and B are represented by feature vectors A = $(x_{1,}x_{2,\ldots\ldots}\ x_n)$ and B = $(y_{1,}y_{2,\ldots\ldots}y_n)$ where n is the dimensionality of the feature space. To calculate the distance between A and B, the normalized Euclidean metric is generally used by

$$dist(A,B) = \surd(\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n})$$

Figure 6: Euclidean Distance

2.4 Unsupervised Learning

In unsupervised learning the training data is unknown. The data is not labeled which means the input variables are given with no corresponding output variable. In unsupervised learning, the algorithms are left to themselves to find unknown pattern and structure. It is useful for less complexity and real-time prediction. K means clustering and hierarchical clustering are some of the popular unsupervised learning examples.

2.4.1 K Means Clustering

K means clustering follow a simple procedure, its goal is to find groups or cluster represent by the variable K. The algorithm works by assigning each data point to a certain cluster based on their features that are given. Each cluster is characterized by its centroid, or center point to labeled new data and they should be placed as far as away from one another to get a better result.

2.4.2 Hierarchical Clustering

Hierarchical clustering is also a process of grouping data, concurrently over a variety of scales of distance, by creating a cluster tree. The tree is not a single set of clusters, as in K-Means, but rather a multi-level hierarchy, where clusters at one level are joined as clusters at the next higher level. This allows determining at what scale or level of clustering is most applicable.

2.5 Evaluation Techniques

To estimate a model accuracy based on randomly partitioning a given data, the following techniques are used:

2.5.1 Cross Validation

Cross validation is a statistical data mining method used to determine the skills and performance of a machine learning model. The dataset is divided and portioned into n folds to use for training and testing. The process is repeated n times for training and testing. It is used commonly to compare and select model as it is easy to implement and altogether have a lower bias. In K-fold cross validation the K represents the number of groups that a given dataset is to be split into [42]. For a specific value of k chosen, if k=10 then it becomes 10-fold cross validation. So, for 10-fold cross validation, the data is divided into 10 parts in a way that each part is about the same size to one another. The process then can be repeated 10 times, with each of the subsamples used only once as the validation data. The results of 10 folds then can be averaged to produce a single estimation.

2.5.2 Confusion Matrix

It is a technique used to describe the performance of a classification model and to determine how well a classifier identify tuples of different classes. It is used to expose the connection between outcomes and predicted class. The above classifier table can be used to predict disease in the way such that if it is:

Confusion Matrix		**Predicted Class**	
		Positive	Negative
Class	Yes	TP	FN
	No	FP	TN

Figure 7: Confusion Matrix Classifier

- True Positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease
- True Negatives (TN): We predicted no, and they don’t have the disease.
- False Positives (FP): We predicted yes, but they don’t actually have the disease.
- False Negatives (FN): We predicted no, but they actually do have the disease

The accuracy of a model can be found by using formula –

$$accuracy = \frac{TP + TN}{P + N}$$

And the error rate –

$$error\ rate = 1 - \ accuracy_{error\ rate}\, \frac{FP + FN}{P\ +\ N}$$

2.5.3 Performance Evaluation

There are two matrices used to determine the performance of data mining which is a recall and false positive rate (FPR).

For a given dataset the recall can be found by using the formula –

$$Recall\ \ = \ \ \frac{True\ Positives}{True\ Positives\ \ +\ False\ Negatives}$$

And false rate –

$$FPR\ \ = \ \ \frac{False\ Positives}{False\ Positives\ \ +\ \ True\ Negatives}$$

The precision of the dataset can be found by –

$$Precision\ \ = \ \ \frac{True\ Positives}{True\ Positives\ +\ \ False\ Positives}$$

2.5.4 Holdout Method

Holdout method is a similar model like cross validation but it is much simpler. The data is randomly portioned. For instance, for a value of k=3, the two third data is trained for model construction and one third for estimating the accuracy [43]. The estimation is less optimistic because only a small part of the data is used to obtain the model.

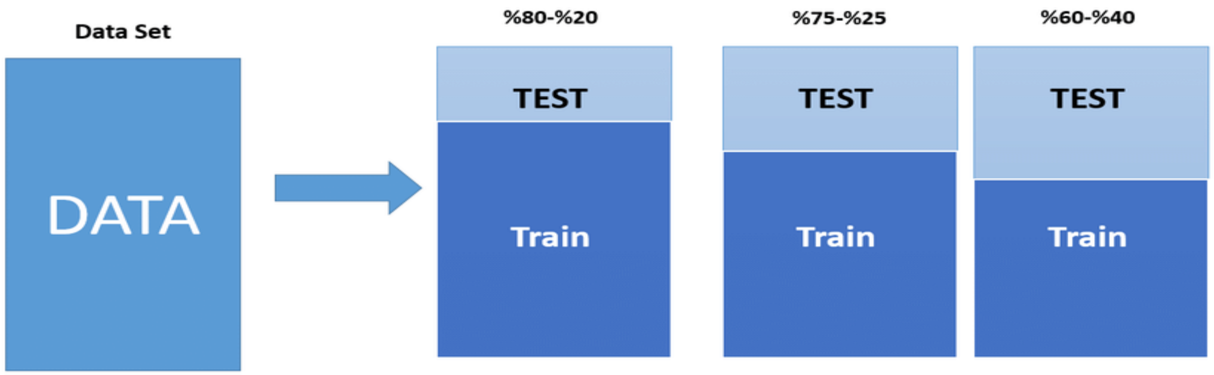


Figure 8: Holdout Method

2.5.5 Bootstrap

Bootstrap is a technique for measuring statistics of the machine learning model. It is a very pessimistic model which works very well with small datasets. During training, each part of the data is selected individually and it is likely to be selected again.

2.6 The Data Mining Tools

2.6.1 WEKA (Waikato Environment for Knowledge Analysis)

WEKA is a machine learning software written in Java, developed at the university if Waikato, New Zealand. WEKA is an open source software available at GNU (General Public License). It contains a collection of visualization tools and algorithm for data analysis and predictive modeling with graphical user interfaces for easy functionality access [44]. WEKA inherits a collection of a machine learning algorithm for solving real-world data mining problems. It supports several standard data mining tasks especially data pre-processing, regression, clustering, classification, visualization, and feature selection. The feature or attribute available in WEKA is of many types, Nominal: One of the predefined list of values, Numeric: A real or integer number, Date, String, Relational. Its key feature is that it is platform independent and open source and consists of various algorithms for data mining and machine learning.

2.6.2 RapidMiner

RapidMiner is another data mining software produced by the company of the same name which provides an integrated environment for data preparation, deep learning, machine learning, text mining, and predictive analytics. RapidMiner utilized for research, education, training, rapid prototyping, and application development and supports all steps of the machine learning process including data preparation, results visualization, validation of model and data optimization [45]. In a study of Blood Research, it is found that RapidMiner provides almost 99 % of an advanced analytical solution through template-based frameworks that speed delivery and reduce errors by nearly eliminating the need to write code. RapidMiner provides data mining and machine learning procedures including - data loading and transformation, data pre-processing visualizing, predictive analytics and statistical modeling, data evaluation and deployment.

2.6.3 R programming

R programming is a language and open source software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The language R is widely used by the statisticians and data miners for developing statistical software and data analysis [46]. R programming was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and right now it is developed by the R Development Core Team (of which Chambers is a member). R and along with its libraries implement a vast variety of statistical and graphical techniques, including linear and non-linear modeling, classical statistical tests, time-series analysis, classification, clustering etc. R programming can be easily extended by functions and extensions, and the community of R is noted for its active contributions. Many of the R’s standard functions are written in R itself, which makes it easy for users to follow the algorithmic choices made.

2.6.4 The Natural Language Toolkit (NLTK)

The Natural Language Toolkit (NLTK) is a leading platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania. It consists of text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning [47].

2.6.5 IBM SPSS Modeler

IBM SPSS Modeler is a powerful, flexible data and text analytics tools that help to build accurate predictive models quickly and intuitively, without programming. It displays patterns and trends in structured and unstructured data easily using unique visual interface supported by advanced analytics. Chosen from a complete range of advanced analytical functions, including state-of-the-art algorithms, automated data preparation and rich, interactive visualization capabilities. It has access to all of IBM SPSS Modeler's predictive capabilities, as well as IBM SPSS Statistics' data transformation, hypothesis testing, and reporting capabilities, from a single interface.

CHAPTER 3

Related Works

In this section different research works that were envisioned to predict the risk of stroke and other diseases.

3.1 STROKE RISK PREDICTION THROUGH NONLINEAR SUPPORT VECTOR CLASSIFICATIONS MODELS [4]

In this research paper, Sabibullah Mohammaed Hanifa and Kasmir Raja S.V presented a study to find the possible risk of stroke by subjecting the risk factors to Support Vector Machines. The authors used the SVM $^{\text{Light}}$ software for implementation. Their data set was extracted from the cohort of population set of various hospital situated at Tiruchirappali city, Tamilnadu, India. They used 100 patient's data with 8 attributes called Hypertension, Diabetes Mellitus, Obesity, Cigarette Smoking, Heart Disease, Prior Stroke, High Cholesterol, and Physical Activity. They used Support Vector Classification model parameters through its kernel function named polynomial kernel and Gaussian (RBF) kernel. The authors evaluated the result through Confusion matrix and show that the rate of the correctness of prediction by RBF is 98% whereby polynomial is 92%. So the author told in this paper that the application of SVM models can be used for the processing of stroke-related risk factor data.

3.2 PREDICTION AND CONTROL OF STROKE BY DATA MINING [5]

This paper described about stroke and the risk factors of stroke. The authors collected 807 data sets within 50 risk factors for stroke by using a standard checklist during the year 2010-2011 in Iran. After pre-processing and cleaning data they have used WEKA Software tool for implementation and data mining techniques such as K-nearest neighbor and C4.5 decision tree for analyzing the data sets. In this work, they have found the performance accuracy of the C4.5 decision tree was 95.42% and the K-nearest neighbor was 94.18%. So they told C4.5 decision tree and K-nearest neighbor can be used for prediction of stroke. However, the authors presented a description of the C4.5 decision tree and K-nearest neighbors only.

3.3 DIABETES PREDICTION USING MEDICAL DATA [6]

In this research paper, the author presented a diabetes prediction system to diagnosis diabetes. They tried to improve the accuracy in diabetes prediction using medical data with various supervised machine learning algorithms namely Naïve Bayes (NB), Multilayer-perceptron (MLP), Random Forest (RF) and the accuracy is noted with different test methods such as 10-fold cross validation (FCV), use percentage split with 66%(PS), and using training dataset (UTD) with pre-processing method and without pre-processing methods. For this work, the author used the WEKA software tool. They used PID dataset and collected dataset with the medical report of 768 persons include 8 features. However, the concluded that the pre-processing methods increase accuracy for Naive Bayes algorithm

3.4 A REVIEW ON PREDICTION OF MULTIPLE DISEASES AND PERFORMANCE ANALYSIS USING DATA MINING AND VISUALIZATION TECHNIQUES [7]

The paper was motivated to construct a basic prototype model which can determine unknown knowledge related to multiple diseases from past database records of specified multiple diseases. In this research work, at first, the authors experimented on the dataset consisting of 1000 records and 14 attributes and the Naive Bayes algorithm gave better accuracy than others. Then the authors experimented again on the dataset of 1000 records within 76 attributes and back-propagation algorithm gave the best accuracy (100%) where the accuracy is given by Naive Bayes algorithms is 90.74%. However, the authors told in their paper that the accuracy result can be changed if we increase the number of datasets and the number of attributes.

3.5 BREAST CANCER CLASSIFICATIONS USING SUPPORT VECTOR MACHINE AND NEURAL NETWORK [8]

In this work, the author tried to carry out using Wisconsin Diagnosis breast Cancer database to classify breast cancer as either benign or malignant. They used dataset consists of 400 observations of patients. Among them 300 are benign and 100 are malignant status. Each instance has 20 features. They used two classes for classifications named cancerous cell and non-cancerous cell and the experiment was carried out by using SVM and Neural Network. The result both SVM and Neural Network were compared on the basis of accuracy and precision. For this experiment, the NN technique is more efficient compared to the SVM technique. However, the author observed that the NN technique is more efficient but the difference is very low.

3.6 DIAGNOSIS OF DIABETES USING CLASSIFICATION MINING TECHNIQUES [9]

The authors have employed two algorithms namely J48 and Naïve Bayes to create the model for diagnosis in determining diabetes in women. They used PID (Pima Indians Diabetes) database of National Institute of Diabetes and Digestive and Kidney Diseases dataset pre-processed in CSV format as their input. The data was divided into a training set and test set by 10-fold cross-validation and percentage split techniques. They pre-processed their dataset by using Weka software tools. The J48 algorithm is used on the dataset using WEKA after which data are divided into "tested positive" or "tested-negative" depending on the final result of the decision tree that is constructed.

3.7 A BRIEF SURVEY ON THE TECHNIQUES USED FOR TECHNIQUES THE DIAGNOSIS OF DIABETES MELLITUS [10]

This paper described about different data mining methods such ask-ford cross-validation and classification, Class wise K Nearest Neighbor, SVM, LDA-Support Vector Machine and Feed Forward Neural Network, Artificial Neural Network, Statistical Normalization, Back Propagation. At first, the authors have been collected 768 cases but after deleting the missing values they had 460 cases for their experiment. They had compared the accuracy of the performance of those classification algorithms and observed that the SVM gave the best accuracy as 81.77% compared to others.

3.8 ASSESSMENT OF STROKE RISK BASED ON MORPHOLOGICAL ULTRASOUND IMAGE ANALYSIS WITH CONFORMAL PREDICTION [11]

Antonis Lambrou, Harris Papadopoulos et.al have been presented a research paper to provide reliable confidence measures for the assessment of stroke by using the Conformal Prediction framework. They evaluated their results of four different conformal predictions respectively: ANN, NB, SVM, and K-NN. For experimentation, the authors applied Principal Component Analysis (PCA) on the dataset and have selected its 6 features which accounted for 98% of its variance. They have used the Leave-One-Out (LOO) method for evaluating. The classifier algorithm ANN-CP was structured with one hidden layer consist of 3 units and the output layer consisting of 2 units. In this paper, the authors observed that SVM classifier has the best accuracy. However, when the authors increase the percentage of confidence the certainty rates start to decrease.

3.9 AN INTEGRATED MACHINE LEARNING APPROACH TO STROKE PREDICTION [12]

Aditya Khosla, Yu Cao et.al. published a paper to present an integrated machine learning approach combining the elements of data imputation, feature selection and prediction. They used different metrics for evaluating their methods such as Notation, Area under the ROC curve (AUC), Concordance Index. They handled their missing data by using mean, median, imputation through linear regression. They evaluated their data imputation quality by using 10-fold cross-validation and used SVM for stroke prediction. The showed in their paper that the combination of Conservative Mean feature selection and Margin-based censored regression gave the best performance.

3.10 A MODEL FOR PREDICTING ISCHEMIC STROKE USING DATA MINING ALGORITHMS [13]

Balar Khalid and Naji Abdelwahab have done a thesis paper to provide a model for predicting Ischemic stroke using data mining algorithms. They have emphasized on prediction and finding risk factors for ischemic stroke so that they could provide a model. They analyzed their data sets through C4.5 DT algorithm and Logistic regression and WEKA 3.6 Software tool. They analyzed their data by using software of Microsoft "XLSTAT". This software based on Visual Basic language. They have found the rates of sensitivity was 77.58% and specificity was 83.03% and error rate was 19.7% with this "XLSTAT" software and AUC (Area Under Curve) area under ROC (Receiver Operating Curve) equals 0.89.

3.11 PREDICTION OF STROKE USING DATA MINING CLASSIFICATION OF STROKE [14]

Vinaytosh Mishra, Dr. Cherian Samuel, Prof. S.K.Sharma3 published a paper to predict diabetes using machine learning. They used Logistic Regression to predict diabetes. In their data, they used Age, Smoking, Parental Diabetes Mellitus, Hypertension & Waist Circumference, Sex, BMI and HBA1C information as the attribute. The data analysis was conducted using the software tool IBM SPSS 20.0. In result, they found the likelihood 78.5565%, Cox & Snell R Square Nagelkerke Square .628, and Nagelkerke R Square 0.839. This research was carried out by Ohoud Almadami and Riyad Alshammari to predict patient at risk of developing stroke by using data mining techniques and to find the patient with who has higher chances to develop stroke. They used three classifier algorithms namely: C4.5, Jrip and multilayers perceptron (MLP). They collected 969 data sets from National Guard hospitals in three different cities in the Kingdom of Saudi Arabia. They collected their data in 2016 from 2nd January to 31st September. They divided their data sets into two classes. First one includes with the patient who has a stroke and the 2nd class includes the patient who has a mimic stroke but they have the diagnosis as they have a stroke. The authors have made a train data set to build their model with 10-fold cross-validation and have made a test data set to evaluate the model. They have used WEKA Software tool for applying their data mining techniques. In this research work it is observed that with the comparison of 10-fold cross validation the Jrip classifier algorithm gave the best performance accuracy (92.60%) but after applying PCA on stroke data C4.5 algorithm gave the best performance on test data set (95.25%).

3.12 LIVER DISEASE PREDICTION USING SVM AND NAÏVE BAYES ALGORITHMS [15]

The author has been used classification algorithms namely Naïve Bayes and SVM for liver diseases prediction in this research work and they implemented their work in MATLAB 2013 tool. The authors described the classification of liver diseases such as cirrhosis, bile duct, chronic hepatitis, liver cancer and acute hepatitis. In this working paper, they showed that SVM algorithms work better than Naïve Bayes algorithms on the basis of performance accuracy. But if we think about execution time, then Naïve Bayes classifier needs minimum execution time.

3.13 APPLICATIONS OF MACHINE LEARNING IN CANCER PREDICTION AND PROGNOSIS [16]

In this paper, the author has described how machine learning works well in cancer prediction and prognosis. They have been shown two histograms. The first one is about the steady increase in published papers using machine learning to predict cancer risk, recurrence and outcome. The another one is about the frequency with which different types of machine methods namely: Naïve Bayes, Genetic algorithm, Fuzzy logic, Clustering, SVM, ANN, Decision Tree are used to predict different types of cancer such as bladder, breast, colorectal, liver, lung, lymphoma, Prostate, Skin, Throat etc. They discussed in their paper about 3 case study which was about the prediction of cancer risk or susceptibility, survivability, recurrence. They worked very well. However, all machine learning studies are not conducted with the same attention by them.

3.14 PREDICTION OF STROKE THROUGH STACKED TOPOLOGY OF ANN MODELS [17]

This paper was motivated to predict the stroke risk by proposing the stacked ANN topology model with higher prediction accuracy. They have collected 300 data for their paper from different hospitals at Tiruchirappalli city, Tamilnadu, India. They analyzed their data by using a back-propagation algorithm and implemented those data through MATLAB 7.3.0-Neural Network Toolbox. They divided their output into three categories namely: High risk, Moderate risk and Low risk. They used the ANN model consist of three layers, respectively input layer, an output layer and one internal layer. They used Confusion matrix as a method for finding the error. They also presented a graphical structure of their predicted and actual outputs of all network. They have been presented a good paper. However, sometimes they made a big difference between their predicted outputs and actual outputs.

Paper Title	Authors	Dataset Information	Used data mining approaches	Performance accuracy
Stroke Risk Prediction through Non-linear Support Vector Classification Models	Sabibullah Mohammaed Hanifa, Kasmir Raja S.V.	100 patient's data with 8 attributes	Gaussian RBF and Polynomial Kernel	98%, 92%
Diabetes Prediction Using Medical Data	DR. D. Asir Antony, Dr.E. Jebamalar, et al.	768 data within 8 attributes and PID dataset.	NB, MLP, RF (WOPP) NB, MLP, RF (WPP)	76.53%, 76.76%, 84.7%, 78.25%, 76.96%, 84.93%
Breast Cancer Classification using Support Vector Machine and Neural Network	Ebrahim Edriss Ebrahim Ali, Wu Zhi Feng	400 data within 20 features	Classification accuracy & precision with SVM and Neural Network	86%,82%,89%,85%, 88%,88%,83%,90%, 92%,88%, 81%, 71%
Prediction and Control of Stroke by Data Mining	Leila Amini et.al	807 datasets with 50 risk factors	C4.5, K-nearest neighbor.	95.42%, 94.18%
Assessment Of Stroke Risk Based on Morphological Ultrasound Image Analysis with Conformal Prediction	Antonis Lambrou, Harris Papadopoulos et.al	Dataset within 6 features	ANN, NB, SVM, and K-NN	71.53%,67.15%,73.72%, 69.34%
Evaluation of the risk of stroke with confidence predictions based on ultrasound carotid image analysis	Antonis Lambrou et.al	274 carotid plaque ultrasound images within 7 different texture features and 2 different morphological features	ANN, SVM, NBC, K-NN with classifier And ANN, SVM, NBC, K-NN with CP	59.90%, 63.50%,54.70%,59.10% 72.26%, 73.72%, 67.52%, 70.80%
Prediction of Stroke Through Stacked Topology of ANN Models	Sabibullah Mohammaed Hanifa, Kasmir Raja S.V.	300 datasets	ST-ANN with low risk, moderate risk and high risk	They presented the predicted output through graphical structure.

Table 1: Comparative Study

CHAPTER 4

4.1 Methodology

The researcher uses some heterogeneous tools for analyzing their dataset and for selecting the appropriate algorithm for the dataset to make the data mining approaches more convenient and faster. There most popular data mining tools are WEKA, MATLAB, R Programming, IBM SPSS Modeler, Rapid Miner, The Natural Toolkit (NLTK) and so on for the context of stroke risk prediction. We have used WEKA Software tool for experimenting our dataset with the configuration of computer system 8GB RAM, Intel®, Core™ i3-7200U CPU 2.71 Processor, Windows 64-bit operating system. The train dataset contains the information of 435 person's information within 15 attributes. We have collected our dataset from the people by using a direct questionnaire form. The attributes of the person such as- age, gender, SBP, DBP, diabetic, ischemic heart disease, race, family history stroke, alcoholic, lack of physical activity, smoking, stress and depression, Statured fat, fibre, Kidney disease. We have proposed a flow chart based on our train dataset. The flow chart of our methodology is given below:

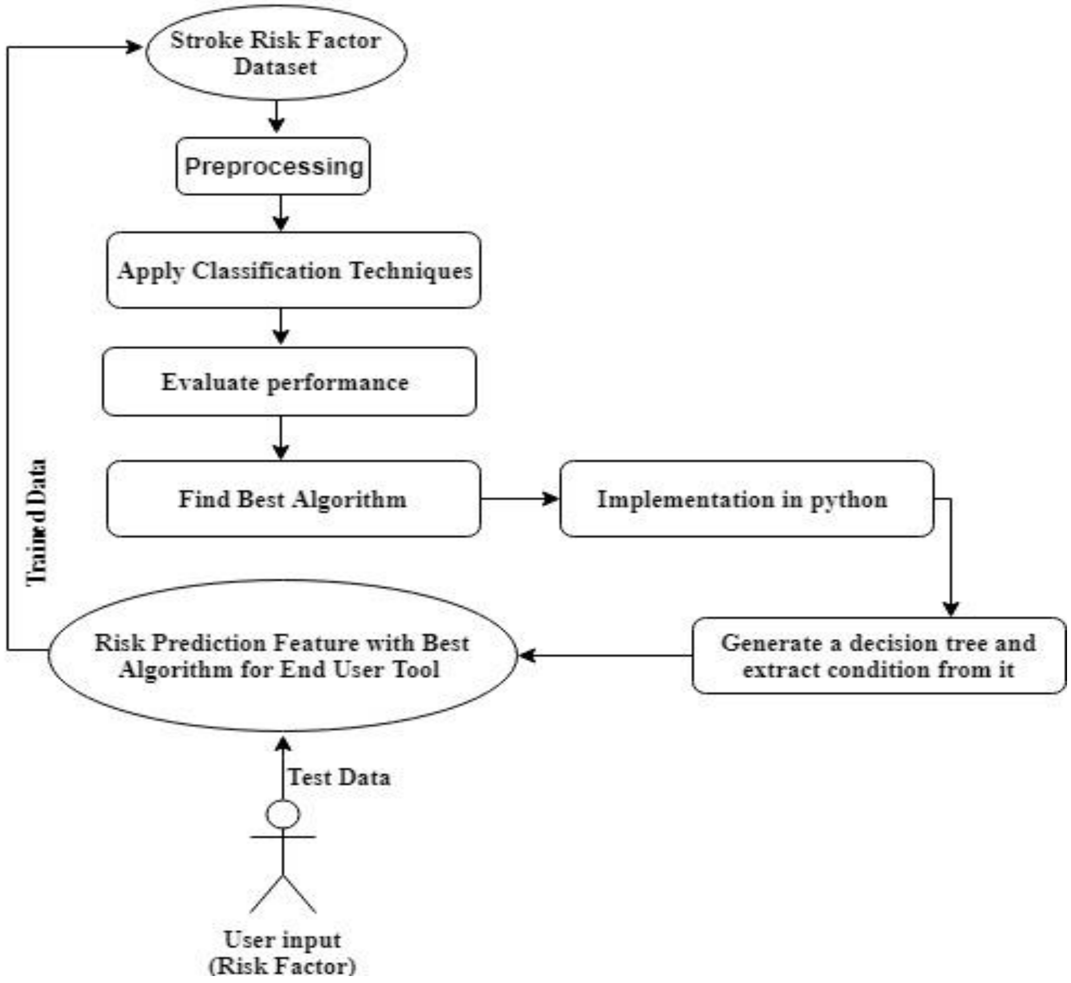

Stroke Risk Factor Dataset
Preprocessing
Apply Classification Techniques
Evaluate performance
Find Best Algorithm
Implementation in python
Generate a decision tree and extract condition from it
Risk Prediction Feature with Best Algorithm for End User Tool
Trained Data
Test Data
User input (Risk Factor)

Figure 9: the flowchart of the methodology.

The algorithm we used in WEKA are conducted keeping some of its properties default and changing some properties to find the best accuracies. In the tables below we are going to represents the properties for Naïve Bayes, J48 decision tree and Classification and regression tree in WEKA that were manipulated while training and testing our dataset.

4.1.1 Naïve Bayes

Naïve Bayes algorithm use to predict the probability of the result or the output for an unseen or unlabeled test input. As it is a type of supervised learning method so, the training data set will available for this algorithm. It is very helpful for large data set and very easy to implement. It works with three step those are prior probability, conditional probability and posterior probability.

First step is **prior probability**. In this step the data set will convert into a frequency table. Second step is **conditional probability**. In this step, a table will generate for each train data depending on the decision class attribute. Third step is **posterior probability** and here if decision class attribute has two labels than prior will count for both class and check the value of which class is greater and finally predict the decision. Suppose we have three attribute class and one decision class. The decision class has two labels those are positive and negative and we also have a test data set which is unseen in the train data set. Naïve Bayes algorithm uses to predict the probability of the result or the output for an unseen or unlabeled test input. There are three equations are given below which represents the classification formula of Naïve Bayes classifier. Here pos and neg represent respectively with the risk of having stroke and without the risk of having stroke.

P(pos) = Total positive risk factors.

P(neg)= Total negative risk factors.

test dataset, $T_{test} = a_1, a_2, a_3 \ldots\ldots\ldots\ldots\ldots\ldots a_n$.

$$P\ (Pos \mid T_{test}) = P\ (a_1|pos) * P\ (a_2|pos) * P\ (a_3|pos) * \ldots\ldots\ldots\ P\ (a_n\ |pos) * P\ (pos) \qquad (1)$$

$$P\ (Neg| \ T_{test}) = P\ (a_1|neg) * P\ (a_2|neg) * P\ (a_3|neg) * \ldots\ldots\ldots\ P\ (a_n\ |neg) * P\ (neg) \qquad (2)$$

$$P\ (\text{ Ttesti } |pos) = \frac{Total\ (\text{Pos} \mid \text{Ttesti})}{\text{Rn}} \qquad (3)$$

Where n is the total number of attributes and i is an incremental number it reaches until the last number of attributes of test dataset and Rn represents the total number of risk factors.

4.1.2 J48 Decision Tree

J48 is one of the most easy and simple algorithms in terms of implementation. It is a most preferable algorithms in supervised learning. At first, it broken down it’s train dataset into smaller subset by using decision tree and at the same time it creates a decision tree which is incrementally developed. J48 decision algorithm is available in WEKA data mining tool. J48 decision tree algorithm is an implementation of java implementation of the C4.5 algorithm. It uses gain ratio as an extension of information gain. By normalizing the information gain using split info, gain ratio can handle the issue of bias. If v is the number of discrete values here then split info and gain ratio can be define by following equation (5) and (6) which are given below:

$$\text{SplitInfo}_A(S) = -\sum_{j=1}^{v} \frac{|\,Sj|}{|\,S|} * \log_2 \frac{|\,Sj|}{|\,S|} \qquad (5)$$

$$\text{GainRatio (A)} = \frac{Gain\ (A)}{\text{SplitInfoA(S)}} \qquad (6)$$

Where, $\frac{|\,Sj|}{|\,S|}$ is weight of the jth partition and the best attribute will be selected with the highest gain ratio.

4.1.3 Classification and regression Tree

In CART algorithm, leaf node represents the output and that is used for predicting where root node represents the input of the dataset. For making prediction with the help of binary CART algorithm is relatively straightforward. If a person wants to create a CART model from the given dataset then he will divide the input space, select the input variable and split point on those variables. This process needs to be done until a suitable tree is being constructed. Recursive binary splitting approach is used for diving the space. CART model used the Gini Index method for creating the split points. It follows the equations (1) to (3) for splitting which are given below:

$$\text{Gini (S)} = 1\text{-}\sum_{i=1}^{m} Pi^2 \qquad (9)$$

Where, S is the sample, Pi is the probability that a tuple in S belongs to class Ci.

If a binary split on attribute A partitions the sample or data S into S1 and S2. The Gini Index of S is:

$$\text{Gini}_A(S) = \frac{|\,S1|}{|\,S|}\text{Gini(S1)} + \frac{|\,S2|}{|\,S|}\text{Gini(S2)} \qquad (10)$$

For selecting the splitting attribute, we need to find the subset which give the minimum value or smaller value of the Gini index. So, we need to follow the equation (11) for finding the attribute with minimum Gini index.

$$\Delta\ \text{Gini (A)= Gini (S)- Gini}_A(S) \qquad (11)$$

Where, A is the attribute with the minimum value of Gini Index is chosen for splitting attribute.

4.2 Mathematical Representation of methodology

In this section we gave a description about our methodology.

Input: Risk factor test dataset.

Output: Predicted risk level, suggestions and tips.

Let,

Risk_algorithm, A;

Performance_accuracy, Acc;

Now,

$$Ttrain = Tn - T\varphi \qquad (12)$$

Where, Ttrain represents train data, Tn represents the total number of instances and T φ represents missing tuple.

$$Acc = A\ (Ttrain, Ttest, Rn) \qquad (13)$$

Where, Ttest represents test data and Rn represents the total number of risk factors.

$$best\_accuracy = \max\ (Acci) \qquad (14)$$

Here, i is the number of algorithms used for prediction and simulation process.

4.3 System Architecture

Our proposed system architecture has been depicted in fig.10, Initially an original dataset including the risk factors of 435 people has been used for selecting the best prediction algorithm. The pre-processing stage was associated following the missing tuple handling method. Then the processed dataset was feed to the database (which will be used as a trained dataset for the end-user tool) and to the classification algorithms for simulation. The performance accuracy has been evaluated using 10-Fold Cross Validation and Percentage Split techniques. Finally, according to the best accuracy, the best algorithm will be chosen for enabling the risk prediction feature of the tool.

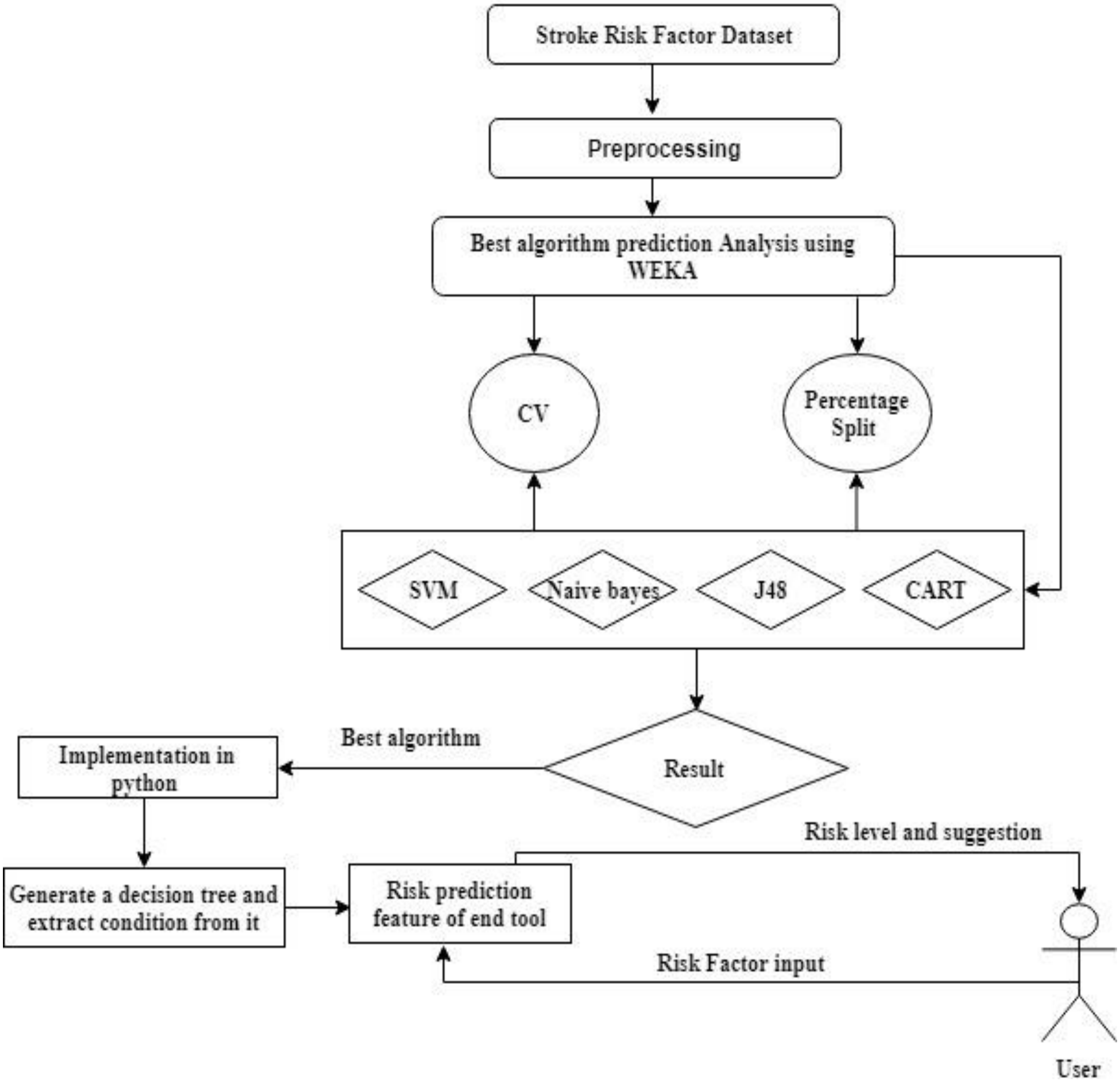

Stroke Risk Factor Dataset
Preprocessing
Best algorithm prediction Analysis using WEKA
CV
Percentage Split
SVM
Naive bayes
J48
CART
Result
Best algorithm
Implementation in python
Generate a decision tree and extract condition from it
Risk prediction feature of end tool
Risk level and suggestion
Risk Factor input
User

Figure 10: Proposed System Architecture

CHAPTER 5

Experimental Analysis

Dataset details and the result analysis is represented in this section.

5.1 Dataset Details

This dataset contains the information of 435 persons. It includes data about peoples including risk factors of developing stroke that may cause stroke. This dataset has been created from a direct questionnaire to people who have recently developed stroke, or who are still not developed the stroke but having few or more risk factors of stroke. The data has been collected from the patients using direct questionnaire from different hospital of Sylhet, Bangladesh. We have collected the information from Sylhet Woman Medical college and Hospital, Jalalabad Ragib Rabeya Medical College and hospital. The description of dataset is given below.

	Number of Attributes	Number of Instances
Risk Factors Dataset	15	435

Table 2: Description of Dataset

Attributes	Values
Age	1.25-34, 2.35-44, 3.45-54,4.55-65,5.65<
Gender	1. Male 2. Female
Systolic BP	1.120>, 2.120-139, 3. 140-160, 4.160<
Diastolic BP	1.180>, 2.80-95, 3.95<
Diabetes	1. No, 2. Yes
Ischemic Heart Disease	1. No, 2. Yes
Family History of stroke	1. No, 2. Yes
Alcoholism	1. No, 2. Yes

Less Physically Active	1. No, 2. Yes
Smoking	1. No, 2. Yes
Stress and depression	1. No, 2. Yes
Saturated Fat↑ ()	1. No, 2. Yes
Fibre↓ ()	1. No, 2. Yes
Chronic Kidney Disease (CKD)	1. No, 2. Yes
Class Attribute	1. Stroke, 2. Non-stroke

Table 3: Description of Attribute

The data pre-processing has been conducted by handling the missing values following the technique of ignoring the tuples with incomplete values. After pre-processing, 435 instances have been remained in total. Among them, 342 are positive values and 93 are negative values. The detail description of the attributes is shown in Table 3. Two class variables are used to find whether the patient is having a risk of developing of stroke (positive) or not (negative).

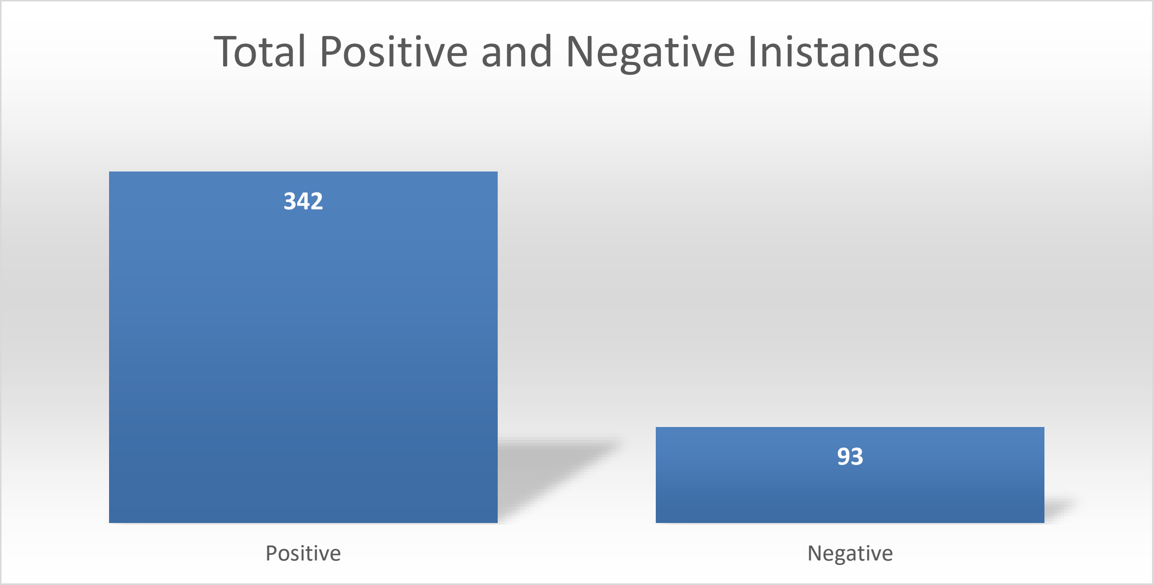


Figure 11: Class Attributes Distribution

5.2 Result Analysis

Performance of different Data Mining techniques on our dataset with detailed accuracy information is represented in the following tables. Although Nave Bayes classifier is one of the most popular algorithms for data prediction, in case of our dataset, the accuracy of it was the lowest for both the Cross-validation method and also for the Percentage split. However, the best result was achieved by using J48 decision tree where using 10-fold cross validation 83.90% instances were classified correctly and using percentage split technique it could classify 80.45% of the instances correctly. In Table 4 to Table 19 we have depicted the detail analysis result. We have found the correctly classified instances and incorrectly classified instances for each algorithm.

	Number of Instances	Percentage
Correctly classified Instances	365	83.90%
Incorrectly Classified Instances	70	16.09%

Table 4: Performance Results from J48 decision tree using (Cross Validation)

	TP Rate	FP Rate	Precision	Recall	F-measure
	0.495	0.067	0.667	0.495	0.568
	0.933	0.505	0.872	0.933	0.901
Weighted Average	0.839	0.412	0.828	0.839	0.83

Table 5: Detailed Accuracy by class from J48 decision tree using 10-fold Cross Validation technique

	Number of Instances	Percentage
Correctly classified Instances	70	80.45%
Incorrectly Classified Instances	17	19.54%

Table 6: Performance Results from J48 decision tree using Percentage Split (80:20)

	TP Rate	FP Rate	Precision	Recall	F-measure
	0.333	0.045	0.7	0.333	0.452
	0.955	0.667	0.818	0.955	0.881
Weighted Average	0.805	0.517	0.79	0.805	0.777

Table 7: Detailed Accuracy by class from J48 decision tree using - Percentage Split

	Number of Instances	Percentage
Correctly classified Instances	345	79.31%
Incorrectly Classified Instances	90	20.68%

Table 8: Performance Results from Naïve Bayes Algorithm using (Cross Validation)

	TP Rate	FP Rate	Precision	Recall	F-measure
	0.548	0.14	0.515	0.548	0.531
	0.86	0.452	0.875	0.86	0.867
Weighted Average	0.793	0.385	0.798	0.793	0.795

Table 9: Detailed Accuracy from by class Naïve Bayes with 10-fold Cross Validation technique

	Number of Instances	Percentage
Correctly classified Instances	69	79.31%
Incorrectly Classified Instances	18	20.68%

Table 10: Performance Results from Naïve Bayes - Percentage Split

	TP Rate	FP Rate	Precision	Recall	F-measure
	0.667	0.167	0.56	0.667	0.609
	0.833	0.333	0.887	0.833	0.859
Weighted Average	0.793	0.293	0.808	0.793	0.799

Table 11: Detailed by class Accuracy from Naïve Bayes– Percentage Split

	Number of Instances	Percentage
Correctly classified Instances	356	81.83%
Incorrectly Classified Instances	79	18.16%

Table 12: Performance Results from Classification Via Regression decision tree Algorithm-10-flod Cross Validation

	TP Rate	FP Rate	Precision	Recall	F-measure
	0.344	0.053	0.64	0.344	0.448
	0.947	0.656	0.842	0.947	0.891
Weighted Average	0.818	0.527	0.798	0.818	0.796

Table 13: Detailed Accuracy from Classification via regression using 10-fold Cross Validation technique

	Number of Instances	Percentage
Correctly classified Instances	71	81.60%
Incorrectly Classified Instances	16	18.39%

Table 14: Performance Results from Classification Via regression using - Percentage Split

	TP Rate	FP Rate	Precision	Recall	F-measure
	0.333	0.03	0.778	0.333	0.467
	0.97	0.667	0.821	0.97	0.889
Weighted Average	0.816	0.513	0.81	0.816	0.787

Table 15: Detailed Accuracy from Classification Via Regression using - Percentage Split

	Number of Instances	Percentage
Correctly classified Instances	342	78.62%
Incorrectly Classified Instances	93	21.37%

Table 16: Performance Results from Support Vector Machine Algorithm – Cross validation

	TP Rate	FP Rate	Precision	Recall	F-measure
	0	0	0	0	0
	1	1	0.786	1	0.88
Weighted Average	0.786	0.786	0.618	0.786	0.692

Table 17: Detailed Accuracy from Support Vector Machine using - Cross Validation

	Number of Instances	Percentage
Correctly classified Instances	66	75.86%
Incorrectly Classified Instances	21	24.13%

Table 18: Performance Results from Support Vector Machine Algorithm – Percentage Split

	TP Rate	FP Rate	Precision	Recall	F-measure
	0	0	0.571	0	0
	1	1	0.759	1	0.863
Weighted Average	0.759	0.759	0.576	0.759	0.654

Table 19: Detailed Accuracy from Support Vector Machine using 10 Percentage Split method

For the more semantic view of the performance of used algorithms using both evaluation techniques are depicted in graphs. In Fig. 12, the performance of the algorithms using Cross-validation evaluation is depicted and in Fig. 13, the results from percentage split have been shown to represent the comparative accuracy of the used algorithms.

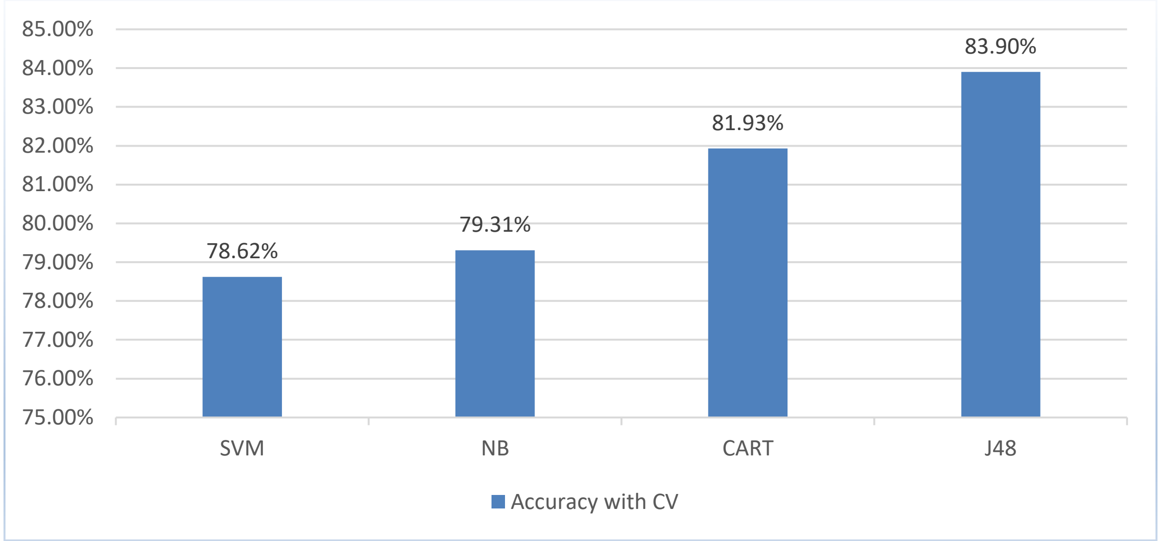


Figure 12: Performance of Classification Algorithms Using Cross-Validation Technique

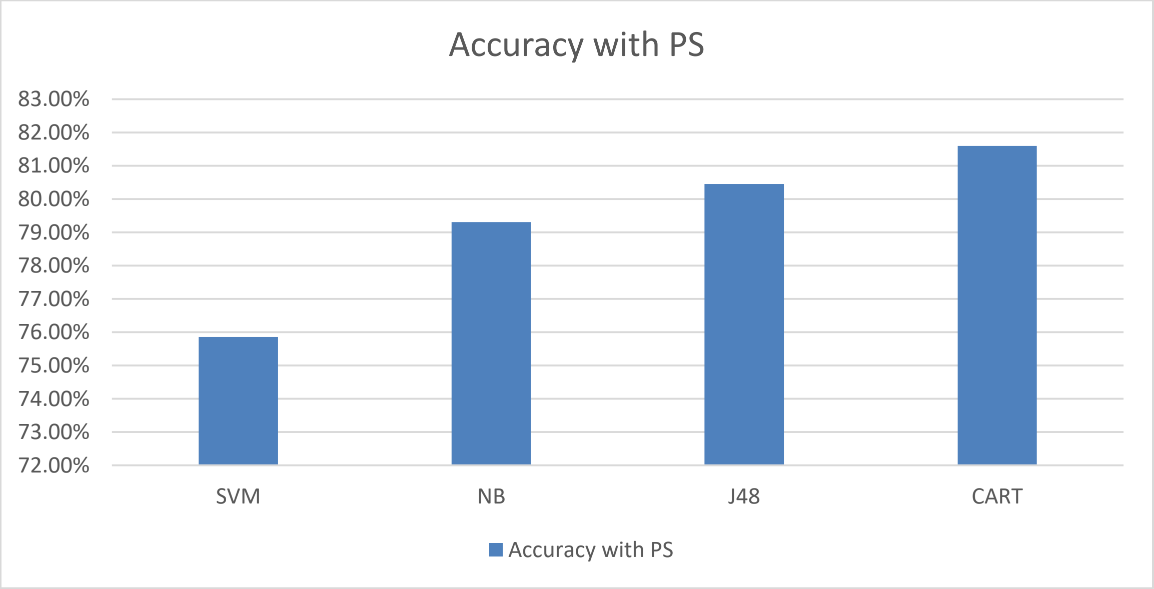
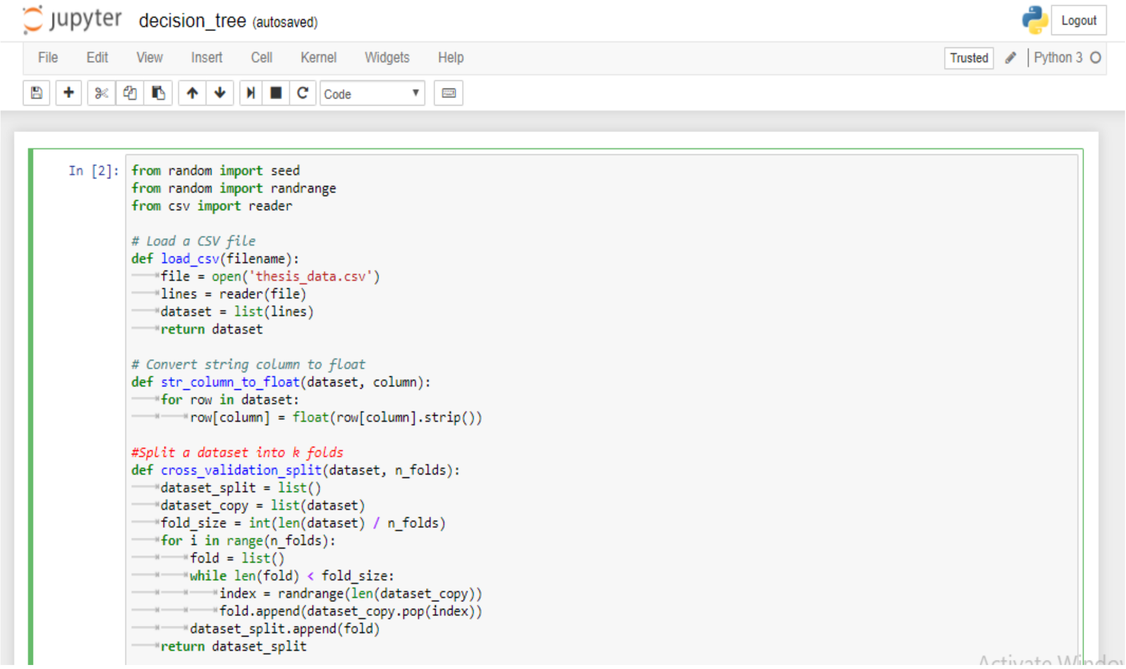


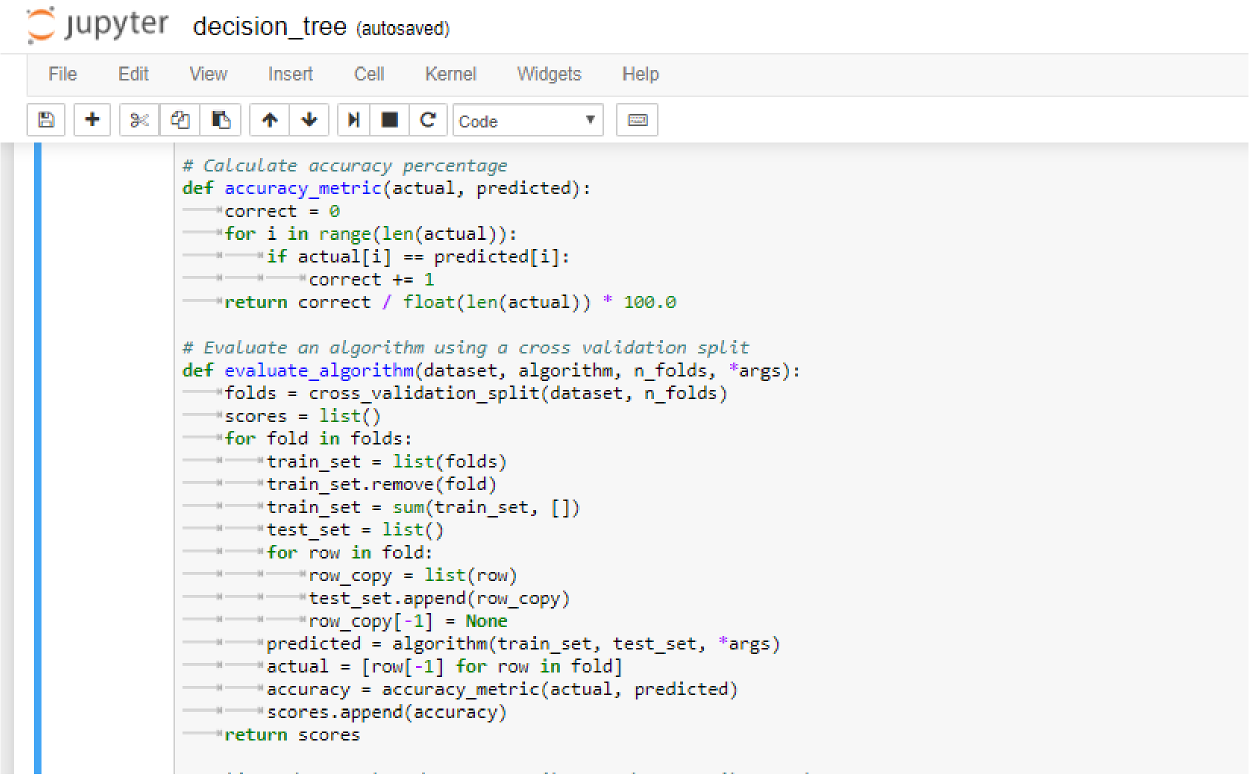
Figure 13: Performance of Classification Algorithms Using Percentage Split Technique

From table 4 to 19 we can see that the decision tree gives the best accuracy performance according to our datasets. After seeing this we have analyzed our datasets by implementing the code of the decision tree in python. We implemented the code in python in two ways. First, we implemented the code with k-fold cross validation. For this we split the dataset into k-fold and calculate the percentage of the performance accuracy. Then we calculate the Gini Index of the attributes and attributes values for the dataset and create a terminal node value. Then we generate a decision tree by following the CART (Classification and Regression Tree) algorithm and find 82.791% the mean accuracy of the algorithm. From figure 14 to 20 all the parts of the code are depicted.



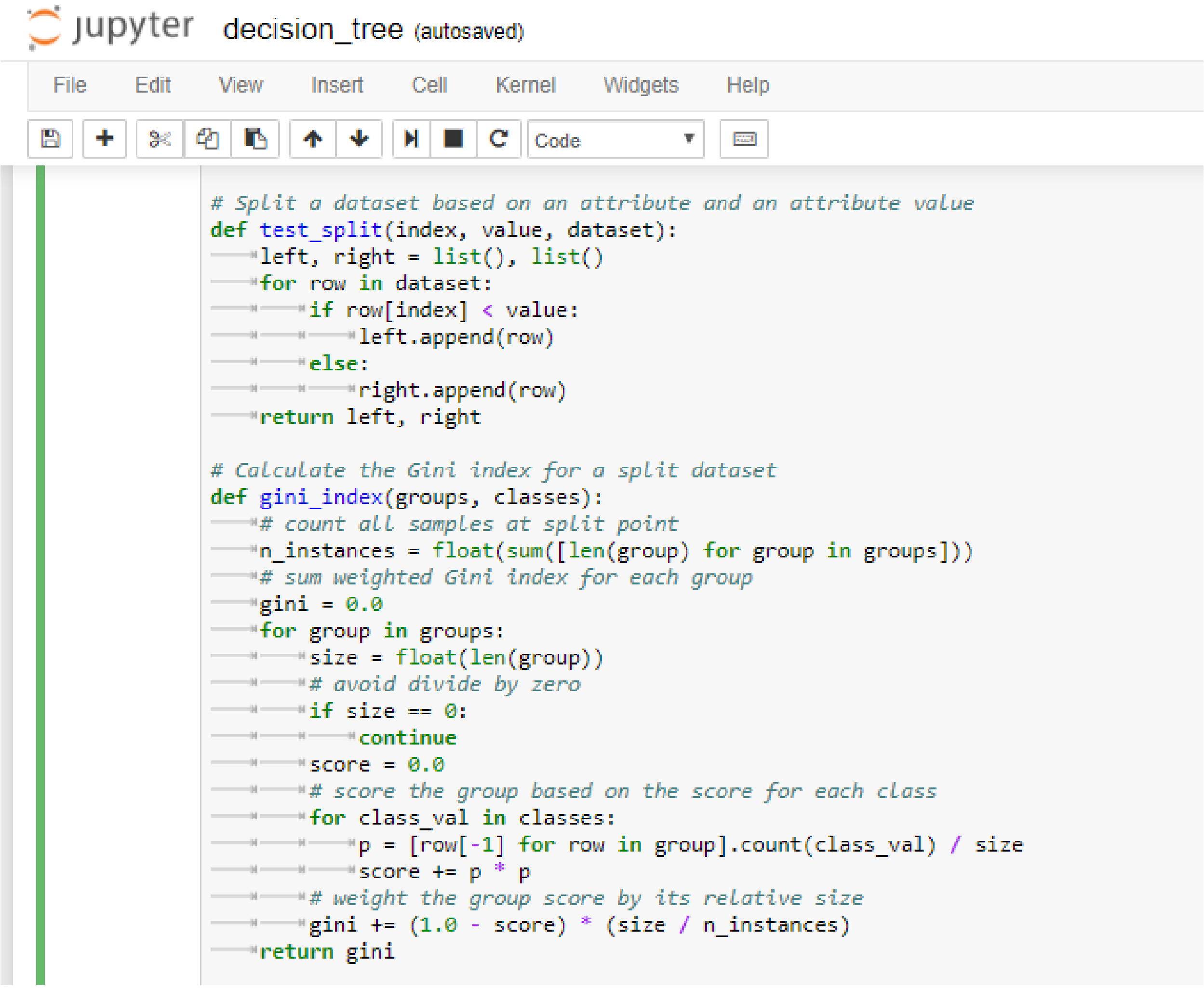
```
from random import seed
from random import randrange
from csv import reader

# Load a CSV file
def load_csv(filename):
    file = open('thesis_data.csv')
    lines = reader(file)
    dataset = list(lines)
    return dataset

# Convert string column to float
def str_column_to_float(dataset, column):
    for row in dataset:
        row[column] = float(row[column].strip())

#Split a dataset into k folds
def cross_validation_split(dataset, n_folds):
    dataset_split = list()
    dataset_copy = list(dataset)
    fold_size = int(len(dataset) / n_folds)
    for i in range(n_folds):
        fold = list()
        while len(fold) < fold_size:
            index = randrange(len(dataset_copy))
            fold.append(dataset_copy.pop(index))
        dataset_split.append(fold)
    return dataset_split
```

Figure 14: load csv file, convert string column to float and split dataset into k-folds.



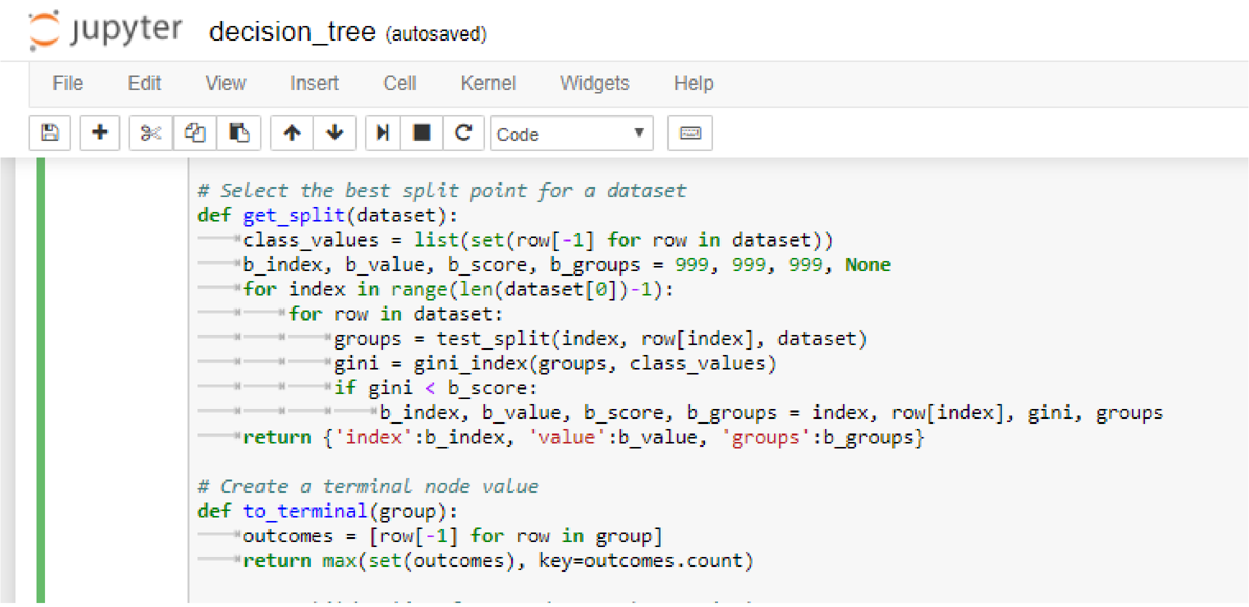
```
# Calculate accuracy percentage
def accuracy_metric(actual, predicted):
    correct = 0
    for i in range(len(actual)):
        if actual[i] == predicted[i]:
            correct += 1
    return correct / float(len(actual)) * 100.0

# Evaluate an algorithm using a cross validation split
def evaluate_algorithm(dataset, algorithm, n_folds, *args):
    folds = cross_validation_split(dataset, n_folds)
    scores = list()
    for fold in folds:
        train_set = list(folds)
        train_set.remove(fold)
        train_set = sum(train_set, [])
        test_set = list()
        for row in fold:
            row_copy = list(row)
            test_set.append(row_copy)
            row_copy[-1] = None
        predicted = algorithm(train_set, test_set, *args)
        actual = [row[-1] for row in fold]
        accuracy = accuracy_metric(actual, predicted)
        scores.append(accuracy)
    return scores
```

Figure 15: calculate the percentage accuracy of performance and evaluate the algorithm.



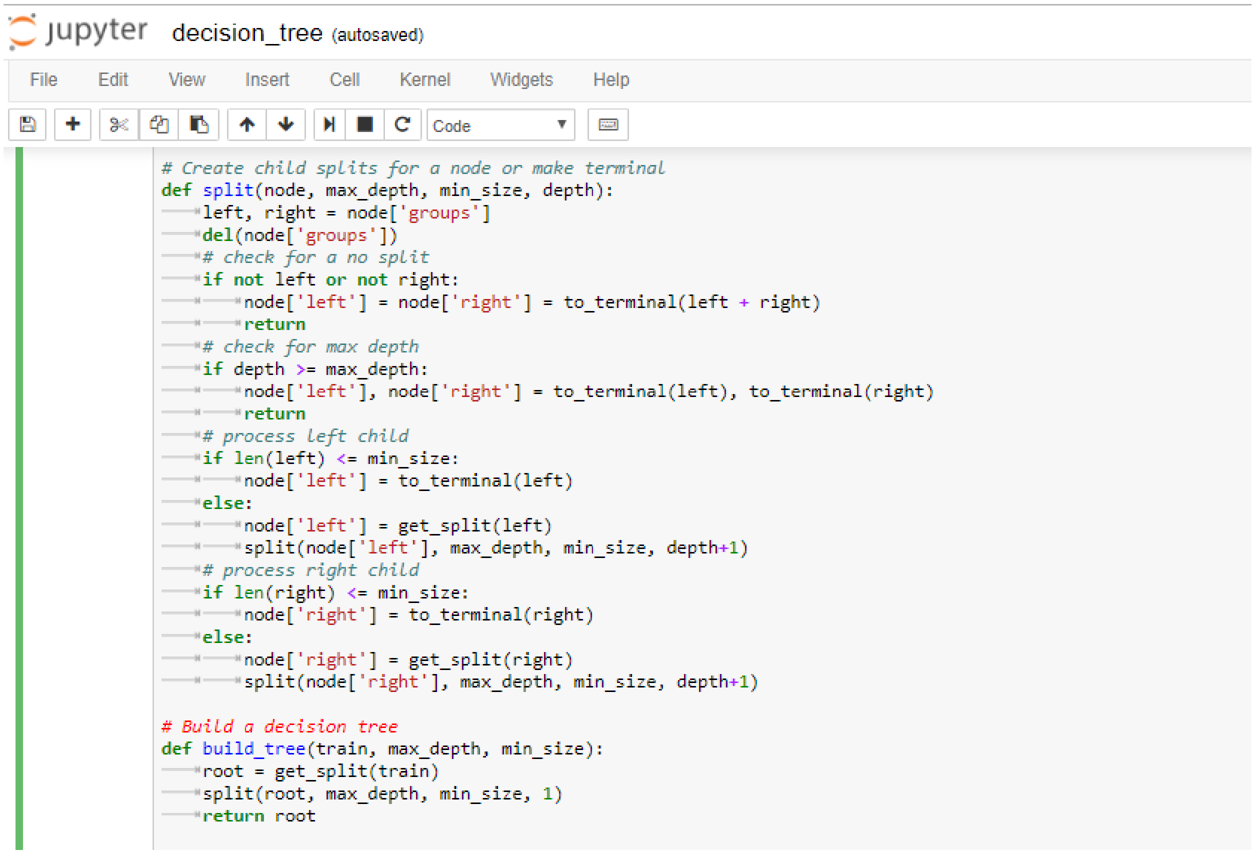
```python
# Split a dataset based on an attribute and an attribute value
def test_split(index, value, dataset):
    left, right = list(), list()
    for row in dataset:
        if row[index] < value:
            left.append(row)
        else:
            right.append(row)
    return left, right

# Calculate the Gini index for a split dataset
def gini_index(groups, classes):
    # count all samples at split point
    n_instances = float(sum([len(group) for group in groups]))
    # sum weighted Gini index for each group
    gini = 0.0
    for group in groups:
        size = float(len(group))
        # avoid divide by zero
        if size == 0:
            continue
        score = 0.0
        # score the group based on the score for each class
        for class_val in classes:
            p = [row[-1] for row in group].count(class_val) / size
            score += p * p
        # weight the group score by its relative size
        gini += (1.0 - score) * (size / n_instances)
    return gini
```

Figure 16: Spilt the dataset based on an attribute and an attribute value and calculate the Gini Index.


jupyter decision_tree (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Code

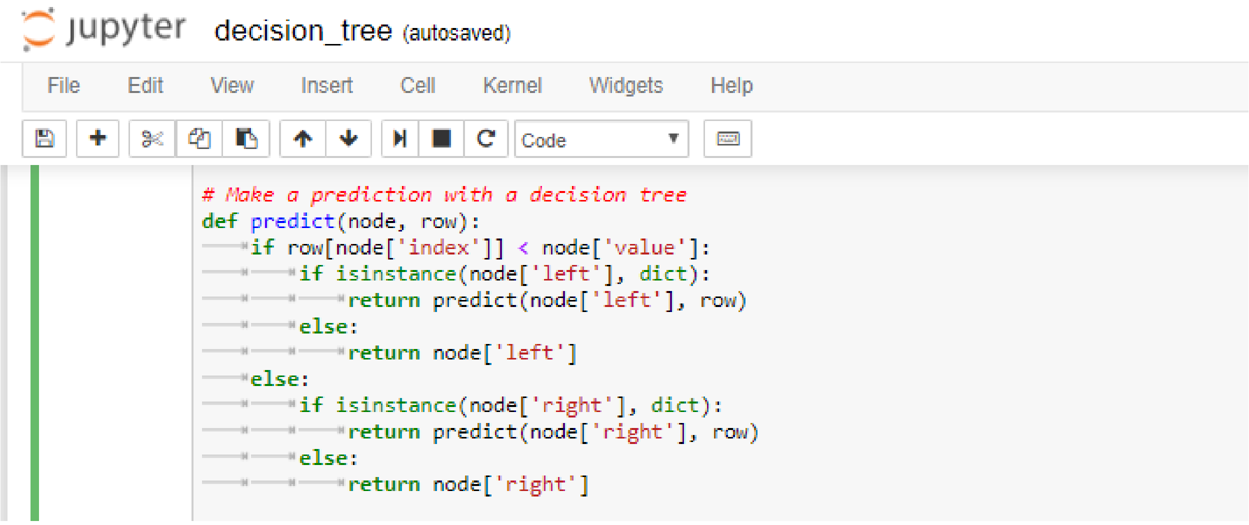
```python
# Select the best split point for a dataset
def get_split(dataset):
    class_values = list(set(row[-1] for row in dataset))
    b_index, b_value, b_score, b_groups = 999, 999, 999, None
    for index in range(len(dataset[0])-1):
        for row in dataset:
            groups = test_split(index, row[index], dataset)
            gini = gini_index(groups, class_values)
            if gini < b_score:
                b_index, b_value, b_score, b_groups = index, row[index], gini, groups
    return {'index':b_index, 'value':b_value, 'groups':b_groups}

# Create a terminal node value
def to_terminal(group):
    outcomes = [row[-1] for row in group]
    return max(set(outcomes), key=outcomes.count)
```

Figure 17: Select best split point and create a terminal node vale.


jupyter decision_tree (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Code

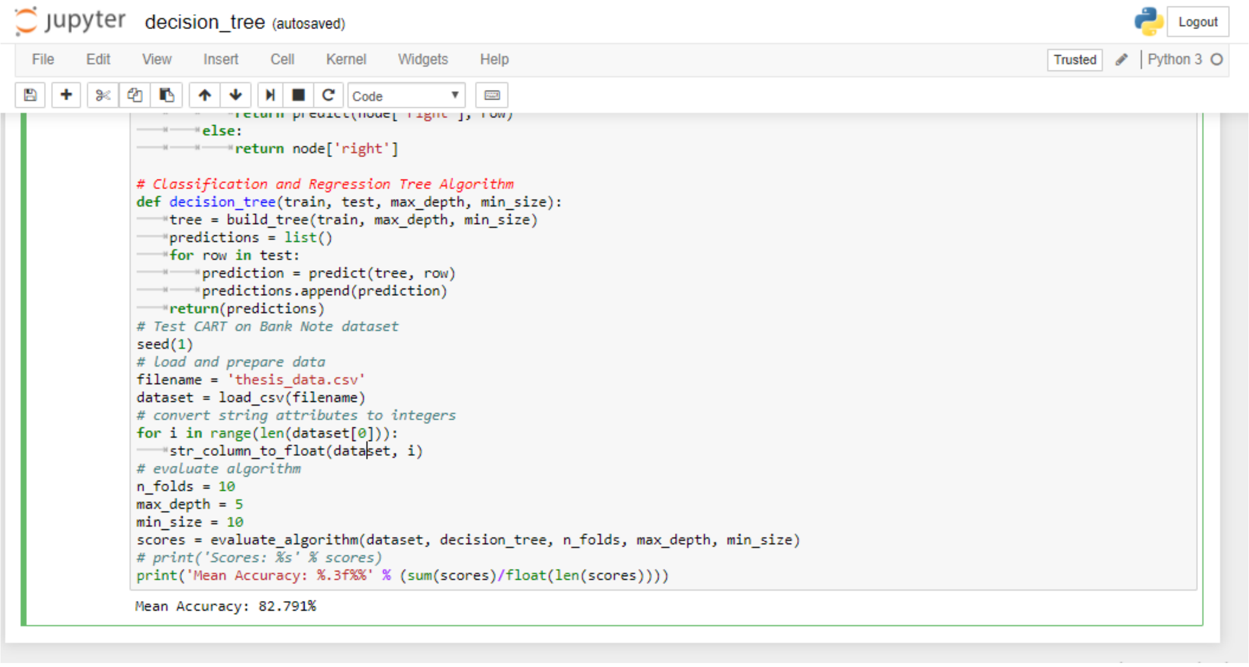
```python
# Create child splits for a node or make terminal
def split(node, max_depth, min_size, depth):
    left, right = node['groups']
    del(node['groups'])
    # check for a no split
    if not left or not right:
        node['left'] = node['right'] = to_terminal(left + right)
        return
    # check for max depth
    if depth >= max_depth:
        node['left'], node['right'] = to_terminal(left), to_terminal(right)
        return
    # process left child
    if len(left) <= min_size:
        node['left'] = to_terminal(left)
    else:
        node['left'] = get_split(left)
        split(node['left'], max_depth, min_size, depth+1)
    # process right child
    if len(right) <= min_size:
        node['right'] = to_terminal(right)
    else:
        node['right'] = get_split(right)
        split(node['right'], max_depth, min_size, depth+1)

# Build a decision tree
def build_tree(train, max_depth, min_size):
    root = get_split(train)
    split(root, max_depth, min_size, 1)
    return root
```

Figure 18: Create child splits for a node and build a decision tree



```python
# Make a prediction with a decision tree
def predict(node, row):
    if row[node['index']] < node['value']:
        if isinstance(node['left'], dict):
            return predict(node['left'], row)
        else:
            return node['left']
    else:
        if isinstance(node['right'], dict):
            return predict(node['right'], row)
        else:
            return node['right']
```

Figure 19: Make a prediction with a decision tree.



```python
            return predict(node['right'], row)
        else:
            return node['right']

# Classification and Regression Tree Algorithm
def decision_tree(train, test, max_depth, min_size):
    tree = build_tree(train, max_depth, min_size)
    predictions = list()
    for row in test:
        prediction = predict(tree, row)
        predictions.append(prediction)
    return(predictions)
# Test CART on Bank Note dataset
seed(1)
# load and prepare data
filename = 'thesis_data.csv'
dataset = load_csv(filename)
# convert string attributes to integers
for i in range(len(dataset[0])):
    str_column_to_float(dataset, i)
# evaluate algorithm
n_folds = 10
max_depth = 5
min_size = 10
scores = evaluate_algorithm(dataset, decision_tree, n_folds, max_depth, min_size)
# print('Scores: %s' % scores)
print('Mean Accuracy: %.3f%%' % (sum(scores)/float(len(scores))))
```

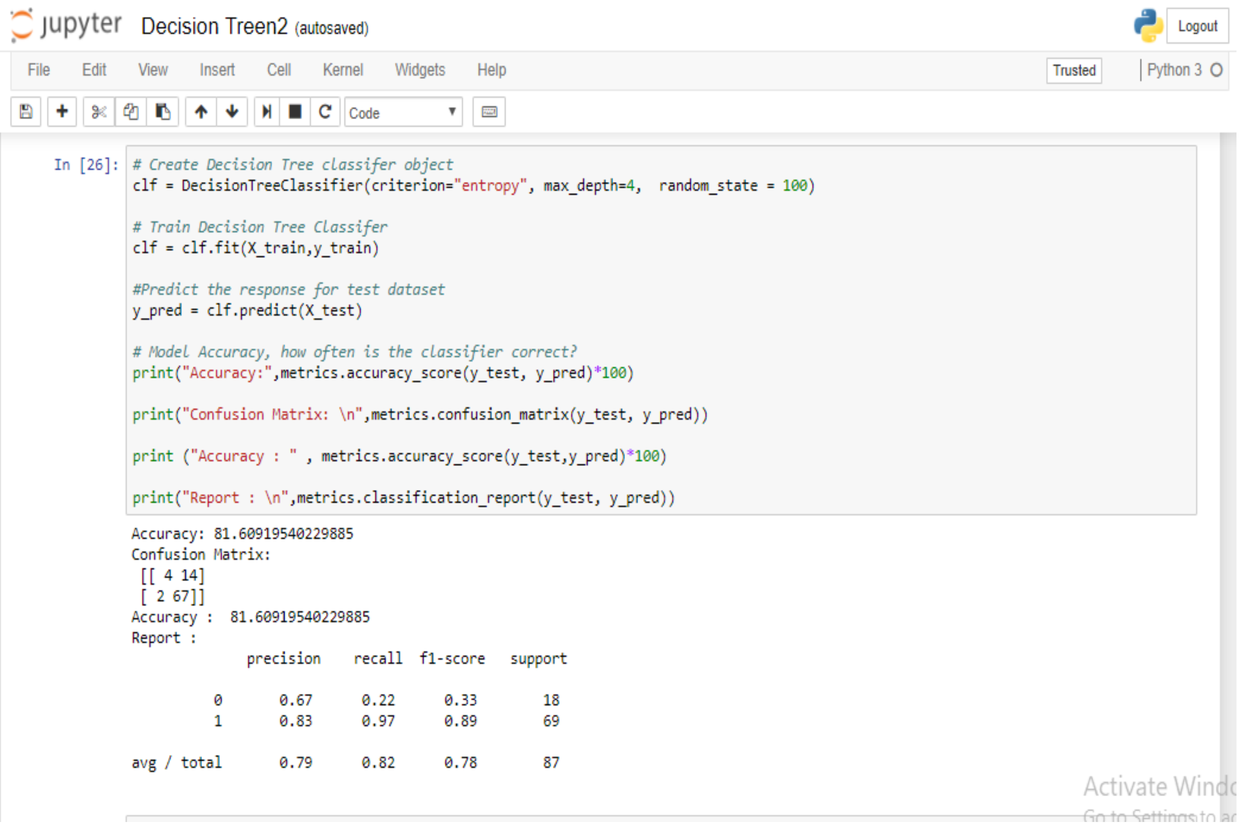
```
Mean Accuracy: 82.791%
```

Figure 20: Apply Classification and Regression Tree algorithm, evaluate the algorithm and find the mean accuracy.

We implemented the code in Python by using library. In figure 21, we showed the screenshot of our code and showed the accuracy, confusion matrix and report of the performance of the algorithm. We found the performance accuracy of the decision tree in python by using entropy which is 81.60% and the report is shown in the table 20.

	Precision	Recall	F1-score	Support
0	0.67	0.22	0.33	18
1	0.83	0.97	0.89	69
avg/total	0.79	0.82	0.78	87

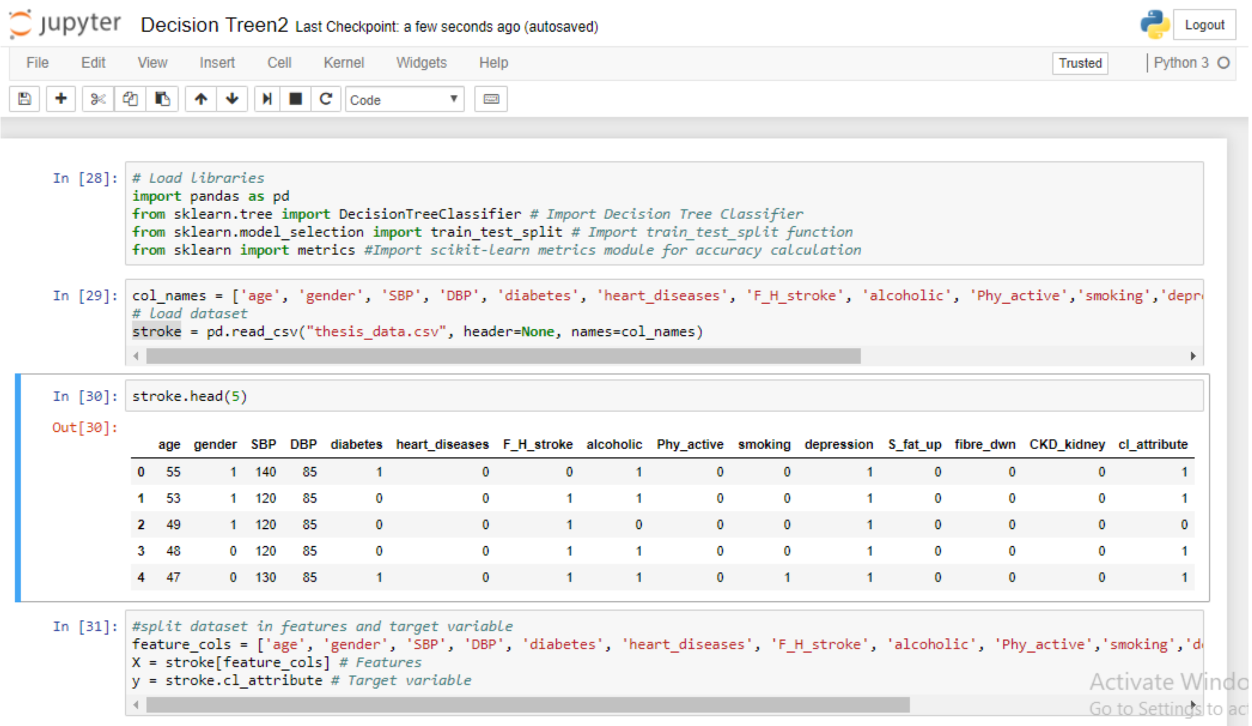
Table 20: The report from the classifier object code of decision tree implemented in python.


jupyter Decision Treen2 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Code
In [26]:
Activate Windo
Go to Settings to ac

```
# Create Decision Tree classifer object
clf = DecisionTreeClassifier(criterion="entropy", max_depth=4,  random_state = 100)

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred)*100)

print("Confusion Matrix: \n",metrics.confusion_matrix(y_test, y_pred))

print ("Accuracy : " , metrics.accuracy_score(y_test,y_pred)*100)

print("Report : \n",metrics.classification_report(y_test, y_pred))
```

```
Accuracy: 81.60919540229885
Confusion Matrix:
 [[ 4 14]
 [ 2 67]]
Accuracy :  81.60919540229885
Report :
              precision    recall  f1-score   support

          0       0.67      0.22      0.33        18
          1       0.83      0.97      0.89        69

avg / total       0.79      0.82      0.78        87
```

Figure 21: Screenshot of the Accuracy, confusion matrix, and report in python by using entropy.



In [28]:
```
# Load libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
```

In [29]:
```
col_names = ['age', 'gender', 'SBP', 'DBP', 'diabetes', 'heart_diseases', 'F_H_stroke', 'alcoholic', 'Phy_active','smoking','depr
# load dataset
stroke = pd.read_csv("thesis_data.csv", header=None, names=col_names)
```

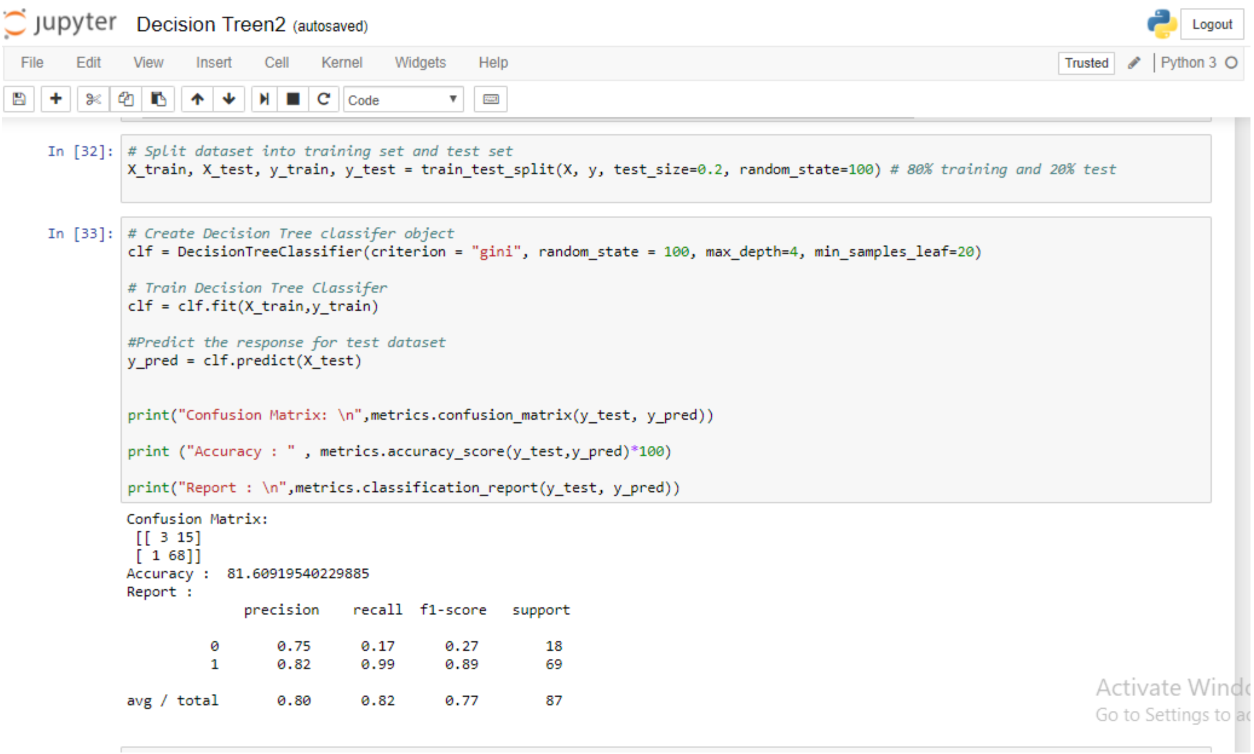
In [30]:
```
stroke.head(5)
```

Out[30]:

	age	gender	SBP	DBP	diabetes	heart_diseases	F_H_stroke	alcoholic	Phy_active	smoking	depression	S_fat_up	fibre_dwn	CKD_kidney	cl_attribute
0	55	1	140	85	1	0	0	1	0	0	1	0	0	0	1
1	53	1	120	85	0	0	1	1	0	0	1	0	0	0	1
2	49	1	120	85	0	0	1	0	0	0	1	0	0	0	0
3	48	0	120	85	0	0	1	1	0	0	1	0	0	0	1
4	47	0	130	85	1	0	1	1	0	1	1	0	0	0	1

In [31]:
```
#split dataset in features and target variable
feature_cols = ['age', 'gender', 'SBP', 'DBP', 'diabetes', 'heart_diseases', 'F_H_stroke', 'alcoholic', 'Phy_active','smoking','d
X = stroke[feature_cols] # Features
y = stroke.cl_attribute # Target variable
```

Figure 22: load csv file, read csv file and showed the dataset of 5 data within all attributes.



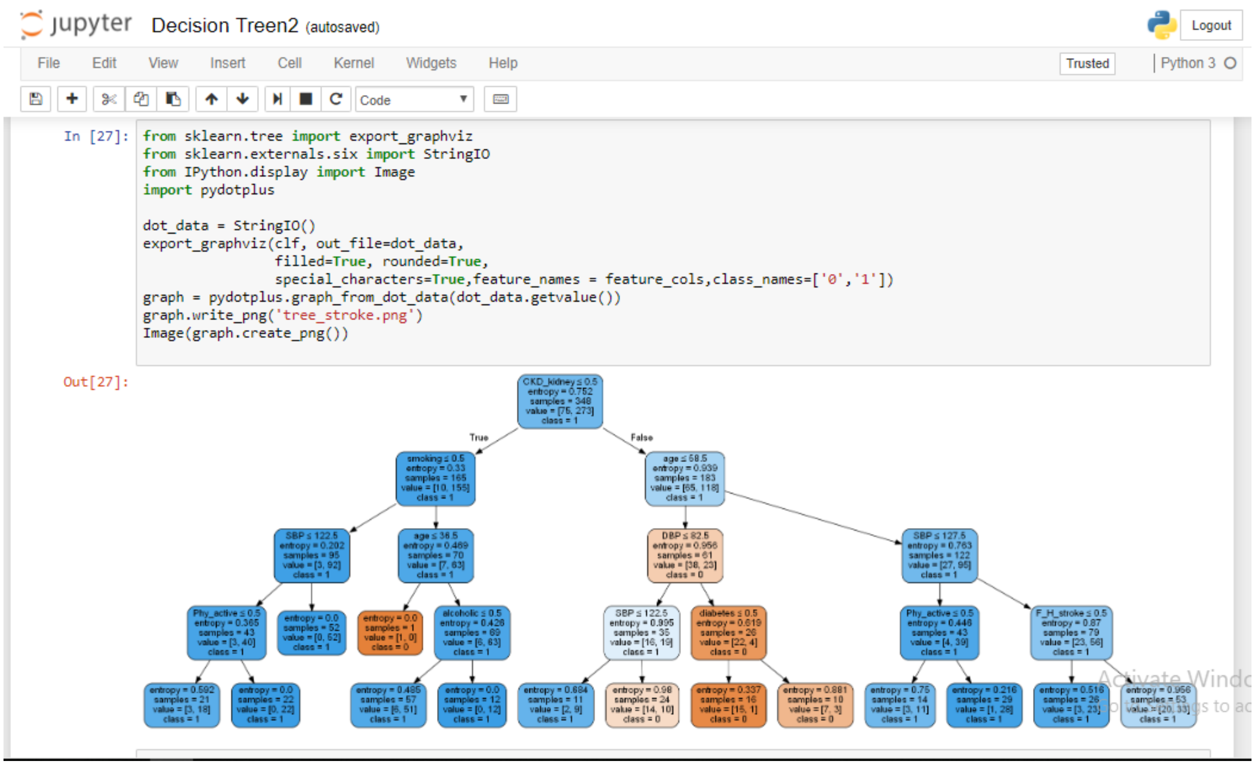
In [32]:
```
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=100) # 80% training and 20% test
```

In [33]:
```
# Create Decision Tree classifer object
clf = DecisionTreeClassifier(criterion = "gini", random_state = 100, max_depth=4, min_samples_leaf=20)

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)


print("Confusion Matrix: \n",metrics.confusion_matrix(y_test, y_pred))

print ("Accuracy : " , metrics.accuracy_score(y_test,y_pred)*100)

print("Report : \n",metrics.classification_report(y_test, y_pred))
```

```
Confusion Matrix:
 [[ 3 15]
 [ 1 68]]
Accuracy :  81.60919540229885
Report :
              precision    recall  f1-score   support

          0       0.75      0.17      0.27        18
          1       0.82      0.99      0.89        69

avg / total       0.80      0.82      0.77        87
```

Figure 23: create decision tree, accuracy of the algorithm, confusion matrix and report of dataset by using Gini Index.

We implemented the code of decision tree in python by using Gini Index which is depicted in figure 22 to 23. We calculated the accuracy of performance which was 81.60%. by using Gini index and then we generate a decision tree.



```
In [27]: from sklearn.tree import export_graphviz
         from sklearn.externals.six import StringIO
         from IPython.display import Image
         import pydotplus

         dot_data = StringIO()
         export_graphviz(clf, out_file=dot_data,
                         filled=True, rounded=True,
                         special_characters=True,feature_names = feature_cols,class_names=['0','1'])
         graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
         graph.write_png('tree_stroke.png')
         Image(graph.create_png())

Out[27]:
```

Figure 24: Generating decision tree from the dataset in python.

We use Scikit-learn’s export_graphviz function for display our decision tree according to our datasets which is shown in figure 24. For plotting the decision tree, we also needed to use pydotplus. Here the export_graphuviz function converted the decision tree classifier into dot file and pydotplus converted this dot file into an image file. Figure 25 shows the plotted decision tree.

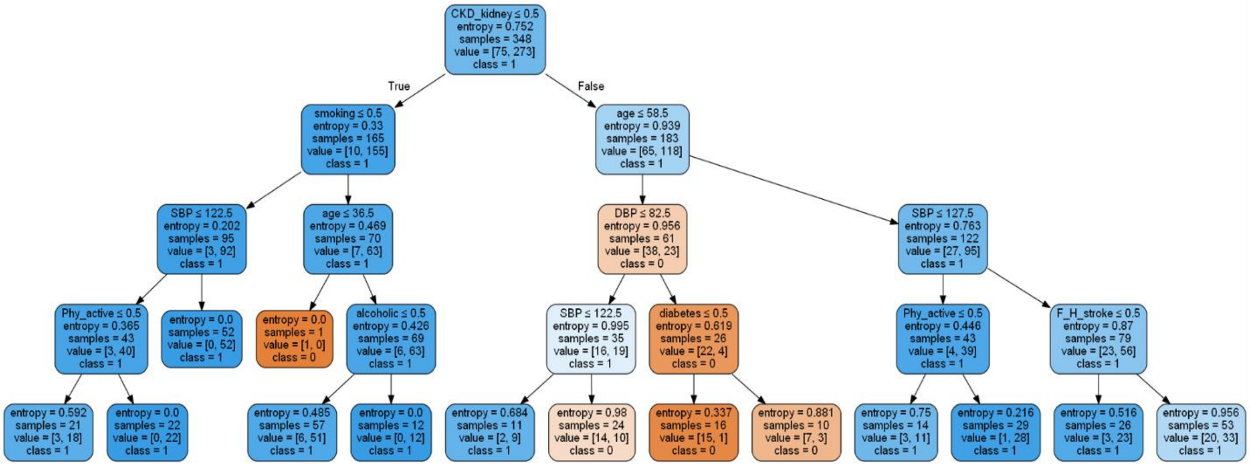
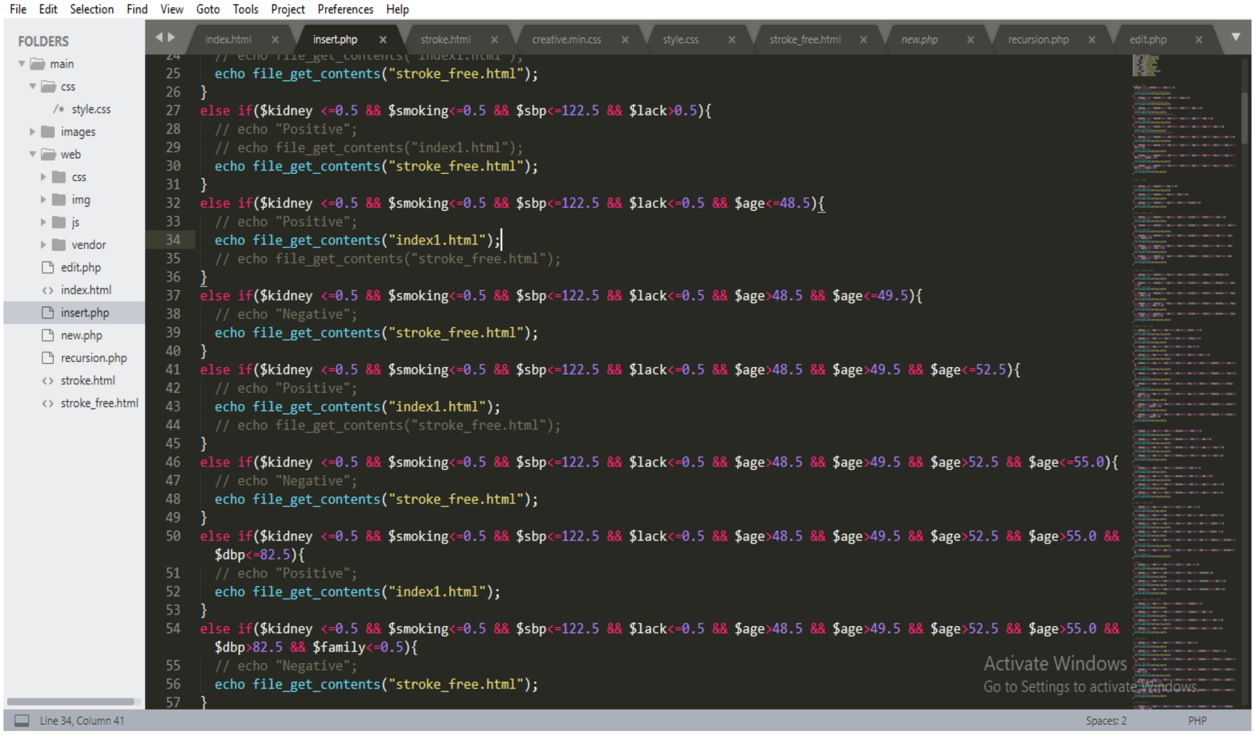


Figure 25: Generated decision tree according to our dataset.

Figure 25 shows the plotted decision tree. There are 1 root node named ckd_kidney and 14 child nodes. We took the logical conditions from this decision tree and implemented the conditions into php for enabling the risk checking feature of our systems. The logical conditions are depicted in figure 26.



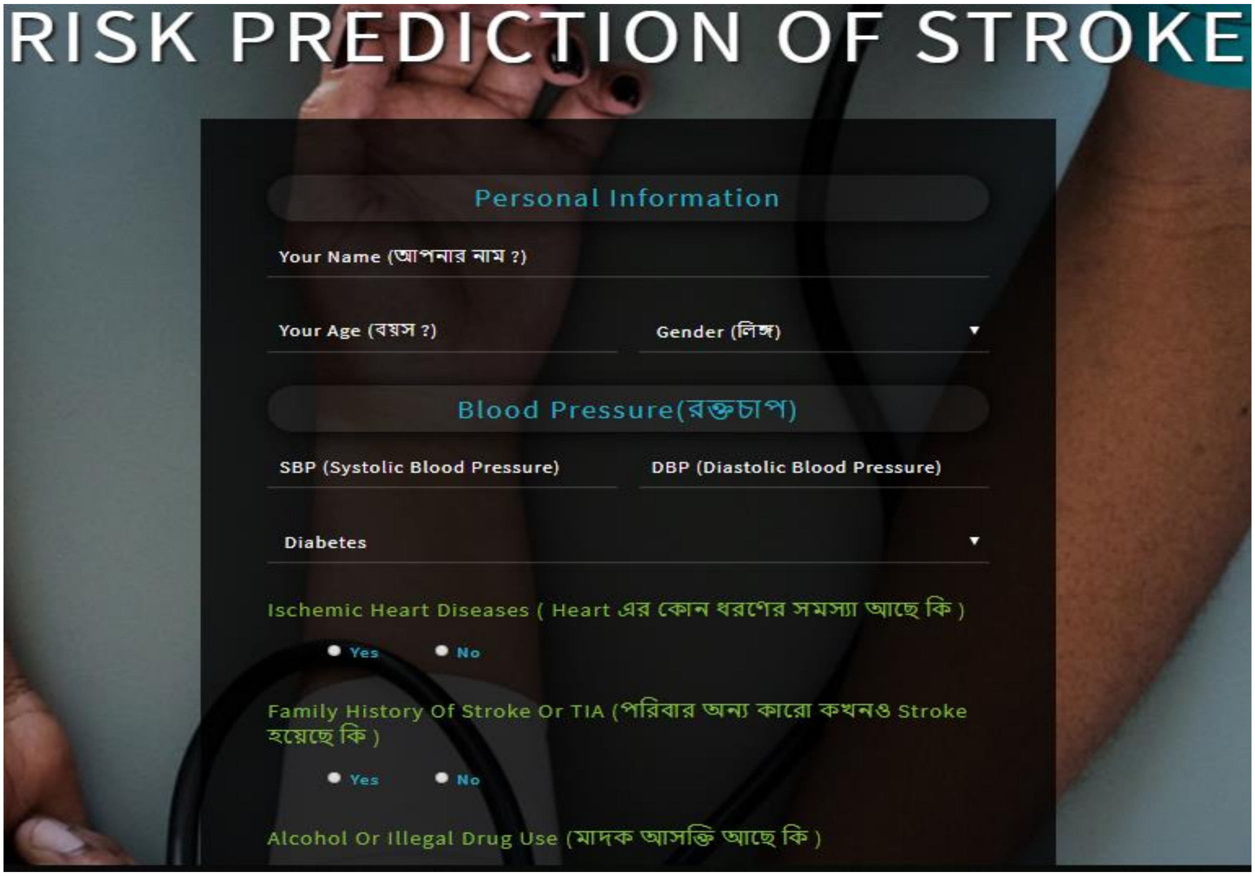
```
  // echo file_get_contents("index1.html");
  echo file_get_contents("stroke_free.html");
}
else if($kidney <=0.5 && $smoking<=0.5 && $sbp<=122.5 && $lack>0.5){
  // echo "Positive";
  // echo file_get_contents("index1.html");
  echo file_get_contents("stroke_free.html");
}
else if($kidney <=0.5 && $smoking<=0.5 && $sbp<=122.5 && $lack<=0.5 && $age<=48.5){
  // echo "Positive";
  echo file_get_contents("index1.html");
  // echo file_get_contents("stroke_free.html");
}
else if($kidney <=0.5 && $smoking<=0.5 && $sbp<=122.5 && $lack<=0.5 && $age>48.5 && $age<=49.5){
  // echo "Negative";
  echo file_get_contents("stroke_free.html");
}
else if($kidney <=0.5 && $smoking<=0.5 && $sbp<=122.5 && $lack<=0.5 && $age>48.5 && $age>49.5 && $age<=52.5){
  // echo "Positive";
  echo file_get_contents("index1.html");
  // echo file_get_contents("stroke_free.html");
}
else if($kidney <=0.5 && $smoking<=0.5 && $sbp<=122.5 && $lack<=0.5 && $age>48.5 && $age>49.5 && $age>52.5 && $age<=55.0){
  // echo "Negative";
  echo file_get_contents("stroke_free.html");
}
else if($kidney <=0.5 && $smoking<=0.5 && $sbp<=122.5 && $lack<=0.5 && $age>48.5 && $age>49.5 && $age>52.5 && $age>55.0 &&
  $dbp<=82.5){
  // echo "Positive";
  echo file_get_contents("index1.html");
}
else if($kidney <=0.5 && $smoking<=0.5 && $sbp<=122.5 && $lack<=0.5 && $age>48.5 && $age>49.5 && $age>52.5 && $age>55.0 &&
  $dbp>82.5 && $family<=0.5){
  // echo "Negative";
  echo file_get_contents("stroke_free.html");
}
```

Figure 26: The logical conditions taken from decision tree and implemented in php for enabling checking feature.

5.3 Proposed Tool for the End Users

It is our motto to provide an easily accessible and effective tool for our end users to make them acquainted with the risk factors of having stroke, so that people can seek medical attention even before developing stroke and reduce the stroke related mortality, morbidity and financial burden. At the same time, making people aware about the risk factors and seeking medical attention timely will delay or even prevent the disease development process as well as prevent further attack of stroke. In this modern era of technology, almost all the people regardless their demography, know the use of websites and web technology. So, we have preferred web technology where a simple website could be beneficial to check the risk of developing stroke by using user's risk factors as input. However, any other regions adopting this idea can change the language according to theirs. This concept was made due to reach mass people of every stage with the contribution of this research work. This website will also provide some useful suggestion and tips to the end users to avoid developing stroke and seek medical attention timely. In the figure 14 a demo input page is given below:

- Can predict the risk of developing stroke in the early stage.
- Can find some healthy tips and helpful suggestions.
- Can be aware before developing stroke.



RISK PREDICTION OF STROKE

Personal Information

Your Name (আপনার নাম ?)

Your Age (বয়স ?)

Gender (লিঙ্গ) ▼

Blood Pressure(রক্তচাপ)

SBP (Systolic Blood Pressure)

DBP (Diastolic Blood Pressure)

Diabetes ▼

Ischemic Heart Diseases (Heart এর কোন ধরণের সমস্যা আছে কি)

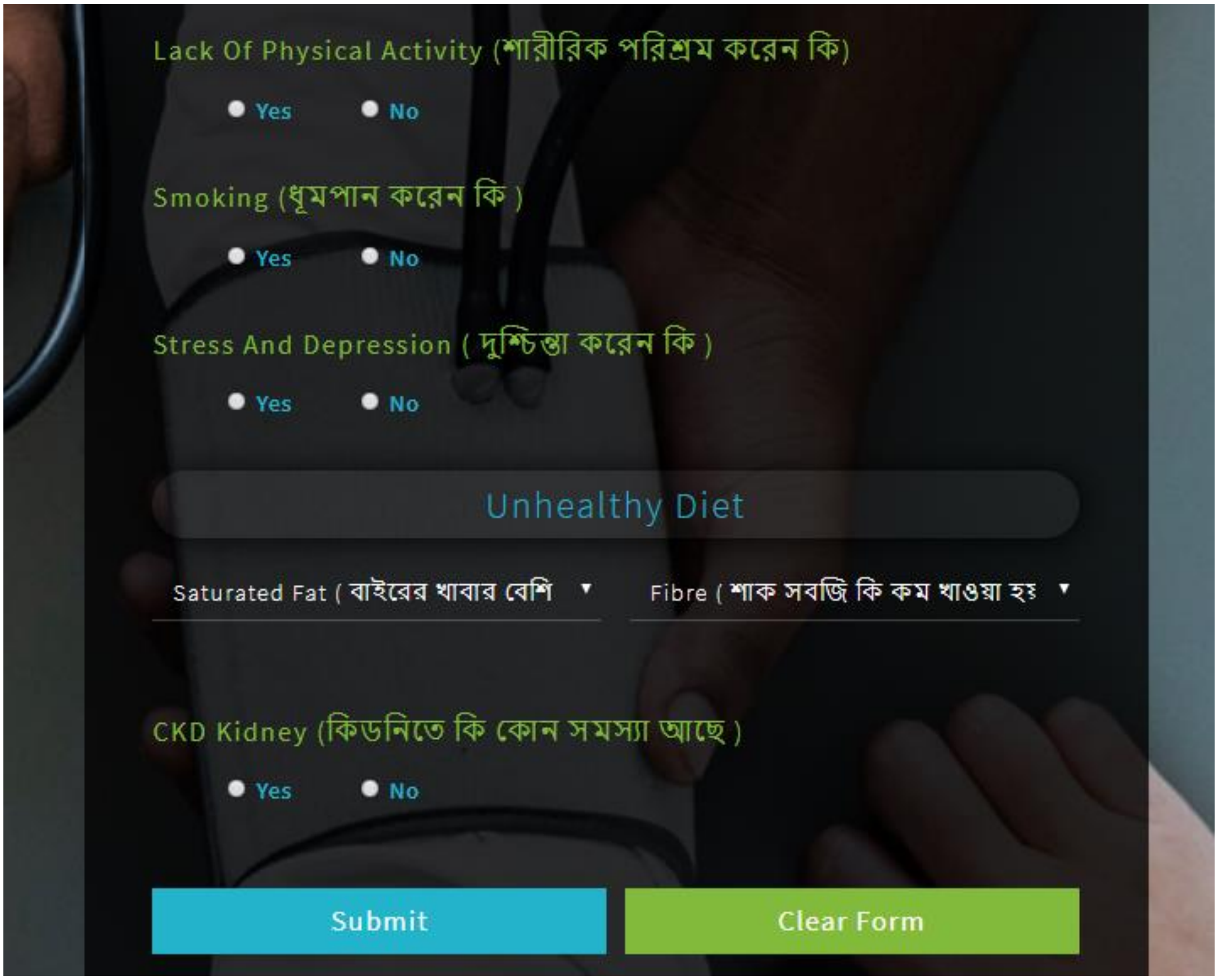
○ Yes ○ No

Family History Of Stroke Or TIA (পরিবার অন্য কারো কখনও Stroke হয়েছে কি)

○ Yes ○ No

Alcohol Or Illegal Drug Use (মাদক আসক্তি আছে কি)

Figure 27: Homepage of the Proposed Website/System



Lack Of Physical Activity (শারীরিক পরিশ্রম করেন কি)

○ Yes ○ No

Smoking (ধূমপান করেন কি)

○ Yes ○ No

Stress And Depression (দুশ্চিন্তা করেন কি)

○ Yes ○ No

Unhealthy Diet

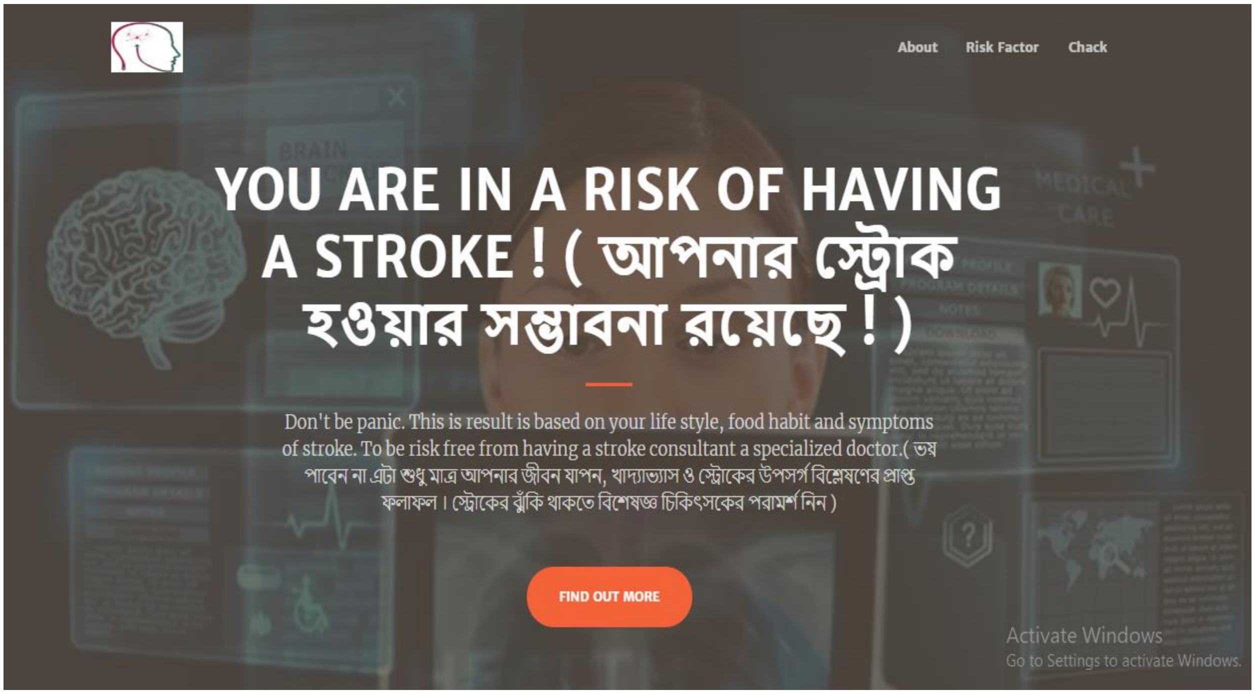
Saturated Fat (বাইরের খাবার বেশি ▾

Fibre (শাক সবজি কি কম খাওয়া হয় ▾

CKD Kidney (কিডনিতে কি কোন সমস্যা আছে)

○ Yes ○ No

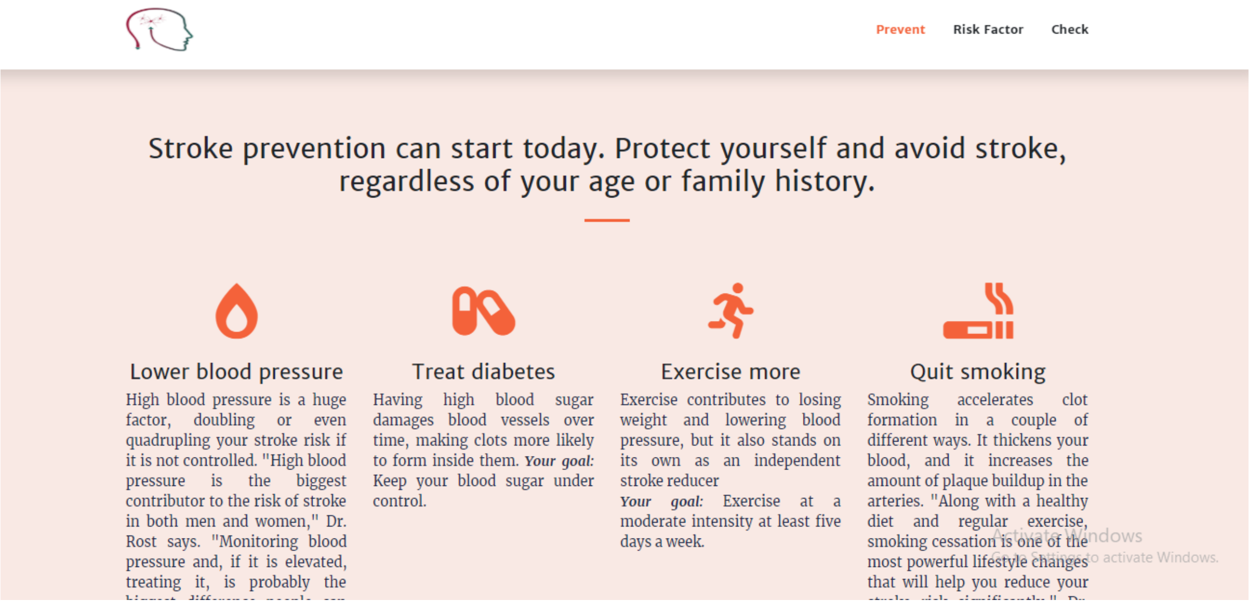
Figure 28: Homepage of the Proposed Website/System



YOU ARE IN A RISK OF HAVING A STROKE ! (আপনার স্ট্রোক হওয়ার সম্ভাবনা রয়েছে !)

Don't be panic. This is result is based on your life style, food habit and symptoms of stroke. To be risk free from having a stroke consultant a specialized doctor.(ভয় পাবেন না এটা শুধু মাত্র আপনার জীবন যাপন, খাদ্যাভ্যাস ও স্ট্রোকের উপসর্গ বিশ্লেষণের প্রাপ্ত ফলাফল । স্ট্রোকের ঝুঁকি থাকতে বিশেষজ্ঞ চিকিৎসকের পরামর্শ নিন)

Figure 29: Risk Checking Page of Stroke



Stroke prevention can start today. Protect yourself and avoid stroke, regardless of your age or family history.

Lower blood pressure

High blood pressure is a huge factor, doubling or even quadrupling your stroke risk if it is not controlled. "High blood pressure is the biggest contributor to the risk of stroke in both men and women," Dr. Rost says. "Monitoring blood pressure and, if it is elevated, treating it, is probably the biggest difference people can

Treat diabetes

Having high blood sugar damages blood vessels over time, making clots more likely to form inside them. ***Your goal:*** Keep your blood sugar under control.

Exercise more

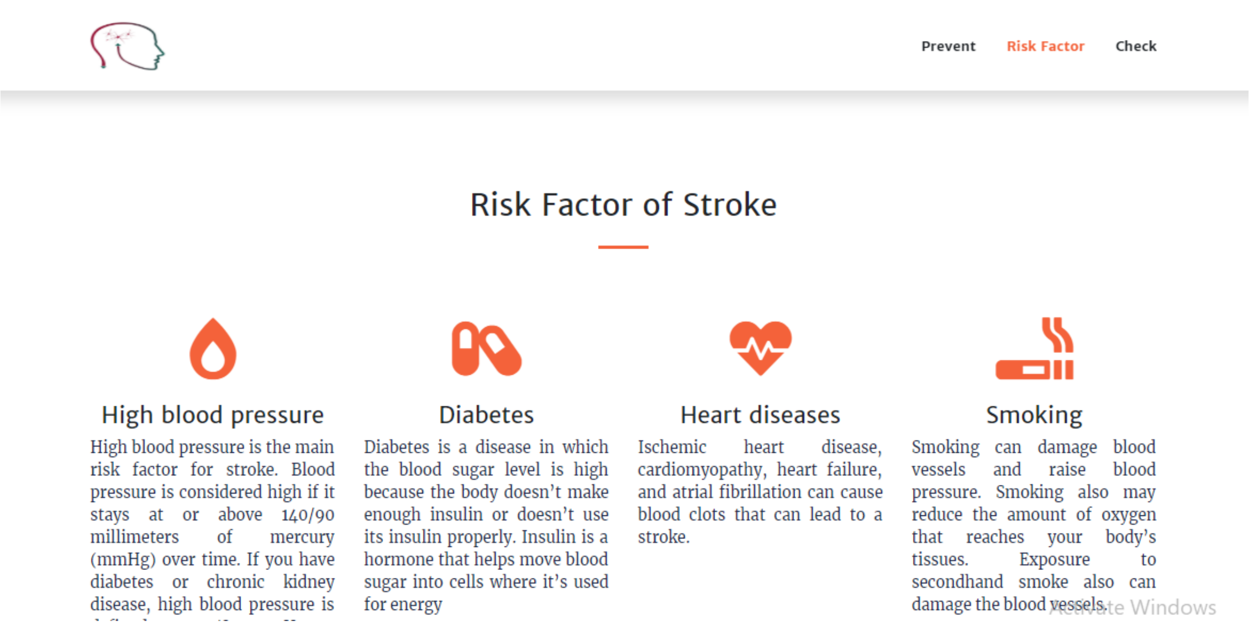
Exercise contributes to losing weight and lowering blood pressure, but it also stands on its own as an independent stroke reducer

Your goal: Exercise at a moderate intensity at least five days a week.

Quit smoking

Smoking accelerates clot formation in a couple of different ways. It thickens your blood, and it increases the amount of plaque buildup in the arteries. "Along with a healthy diet and regular exercise, smoking cessation is one of the most powerful lifestyle changes that will help you reduce your stroke risk significantly," Dr.

Figure 30: Risk Checking Page of Stroke



Risk Factor of Stroke

High blood pressure

High blood pressure is the main risk factor for stroke. Blood pressure is considered high if it stays at or above 140/90 millimeters of mercury (mmHg) over time. If you have diabetes or chronic kidney disease, high blood pressure is

Diabetes

Diabetes is a disease in which the blood sugar level is high because the body doesn’t make enough insulin or doesn’t use its insulin properly. Insulin is a hormone that helps move blood sugar into cells where it’s used for energy

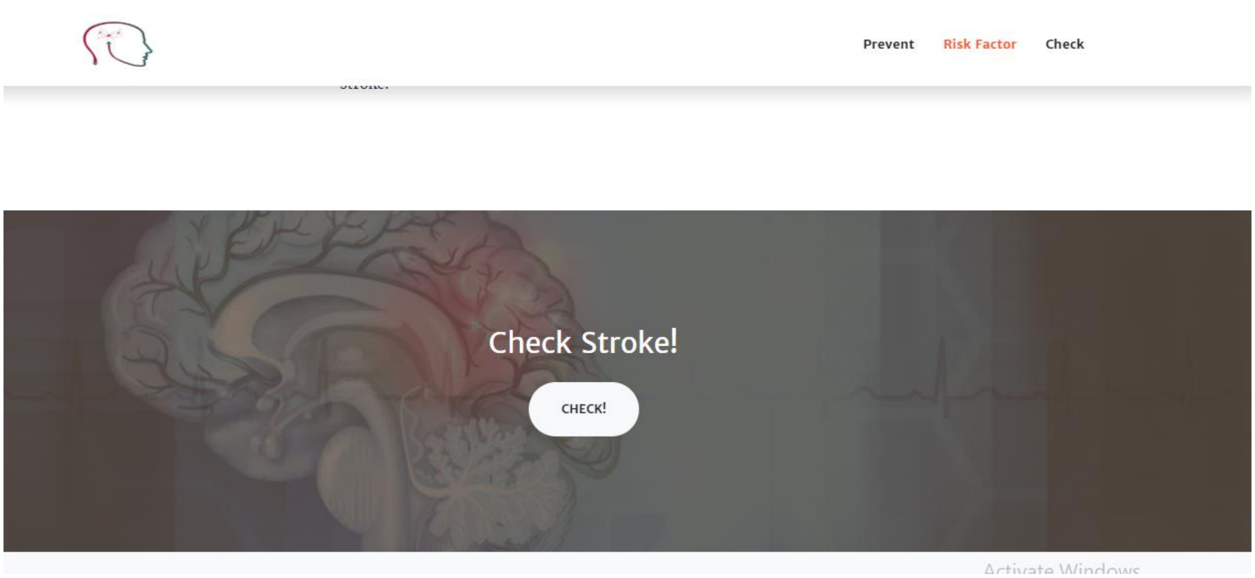
Heart diseases

Ischemic heart disease, cardiomyopathy, heart failure, and atrial fibrillation can cause blood clots that can lead to a stroke.

Smoking

Smoking can damage blood vessels and raise blood pressure. Smoking also may reduce the amount of oxygen that reaches your body’s tissues. Exposure to secondhand smoke also can damage the blood vessels.

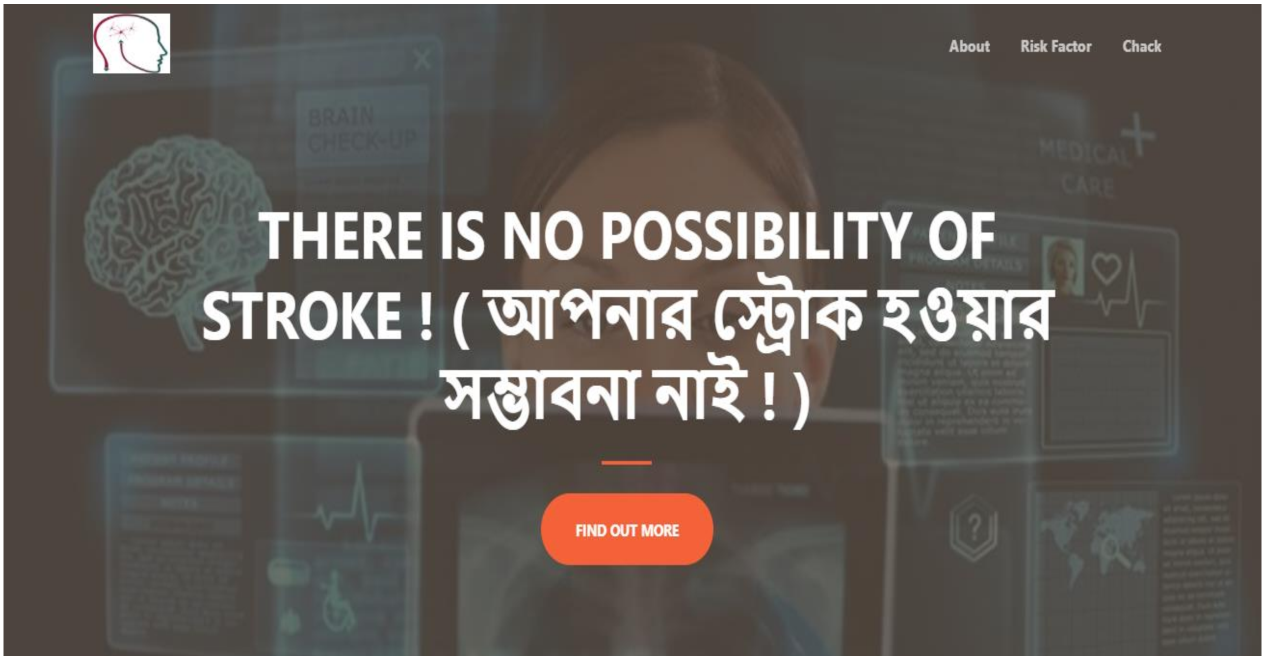
Figure 31: Risk Checking Page of Stroke



stroke.

Check Stroke!

Figure 32: Risk Checking Page of Stroke



THERE IS NO POSSIBILITY OF STROKE ! (আপনার স্ট্রোক হওয়ার সম্ভাবনা নাই !)

Figure 33: Risk Checking Page of Stroke

CHAPTER 6

Conclusion

All the statistics are showing that the global prevalence of stroke is rising where people are still unaware of the risk factors of developing stroke. Knowing the risk factors by any means could make them alert to reduce the incidence of stroke and its aftermaths effectively. This research paper presented a system for risk prediction of stroke by using the different data mining techniques. As data mining techniques have some algorithms for predicting many diseases so we used some algorithms of data mining techniques namely SVM, NB, J48, CART, MLP and so on.[11] We observed in this work that the J48 algorithm gave the best accuracy. We also provide a tool for the end users so that they can know the risk level of their having stroke. With the help of this tool, we can increase awareness about stroke. However, we have collected only 435 data as our train dataset, it can be updated by increasing the number of instances and can be implemented in others data mining techniques for prediction purpose.

6.1 Limitations

- As it is a newly created dataset within the risk factors only so it is quite small still.
- Since we collected our datasets from the different hospitals in Sylhet, it was hard to find the stroke patients. So, if anyone can collect the data from outside hospitals in Sylhet then the effectiveness of the dataset will increase further.

6.2 Challenges

As our research is based on a lots of data, though our main target or first priority and as expected this task wasn't too easy. We knocked some famous doctor's door regarding our thesis and finally after many we can convince Dr. Mahjuba Umme Salam. We discussed on it and she explained some medical terms of stroke and then we study about this terms.

After that we went to many hospital in Sylhet and collected data from Jalalabad Ragib Rabea hospital, Sylhet women hospital, North East Medical College, Sylhet. As we all know the doctor's schedule is for busy so, we had to communicate more informally day and night. We also tried to contact with the patients and collect information about their lifestyle, food habits family history and it was also tough task. We had to consider their medical condition, mood and collect data. For a smart and efficient work we create a questionnaire form based on precautionary question of stroke and tried to relate the cause of stroke happened. Simplifying the questionnaire form. (Like the form has some medical team many of them people can't understand so we had to simplify them in many case this needs explanation too)

We collect more than 400 patient's information and researched their lifestyle. Find out most common factors and made another form for the public where people have to answer few question and we will reply them is they are in risk or not.

6.3 Future Work

- Developing a website or mobile apps developed using J48 method for mass people can be considered as the future scope of this work.
- Regularly updated data from the user in the system will enrich the dataset.
- Analysis of real system performance can also be conducted.

References

1. O. O. ,. M. O. &. S. S. Walter Johnson, "World Health Organization," 2016. [Online]. Available: https://www.who.int/bulletin/volumes/94/9/16-181636/en/. [Accessed 3 January 2019].
2. W. S. Organization, "World Stroke Organization," [Online]. Available: https://www.world-stroke.org/component/content/article/16-forpatients/84-facts-and-figures-about-stroke.
3. A. L. D. e. a. Chandratheva, "Population-Based Study of Behavior Immediately After Transient Ischemic Attack and Minor Stroke in 1000 Consecutive Patients," *Lessons for Public Education,* p. 41, 2010.
4. K. R. S. Sabibullah Mohamed Hanifa, "Stroke Risk Prediction through Non-linear Support Vector Classification Models," *International Journal of advanced Research in Computer science,* vol. 1, pp. 1-8, 2010.
5. D. E. J. L. B. S. B. Dr. D. Asir Antony Gnana Singh, "Diabetes Prediction Using Medical Data," *Journal of Computational Intelligence in Bioinformatics,* vol. 10, pp. 1-8, 2017.
6. H. S. N. F. S. Ajinkya Kunjir, "A Review on Prediction of Multiple Diseases and Performance Analysis using Data Mining and Visualization Techniques," *International Journal of Computer Applications (0975 – 8887),* vol. 155, pp. 35-38, 2016.
7. W. Z. F. Ebrahim Edriss Ebrahim Ali, "Breast Cancer Classification using Support Vector Machine and Neural Network," *International Journal of Science and Research (IJSR),* vol. 5, no. 3, pp. 1-6, 2013.
8. S. J. R. S. Aiswarya Iyer, "Diagnosis of diabetes using classification mining techniques," *International Journal of Data Mining & Knowledge Management Process (IJDKP),* vol. 5, pp. 1-14, 2015.
9. D. S. W. Joseph A. Cruz, "Applications of Machine Learning in Cancer Prediction and Prognosis," *SAGE Journals,* pp. 1-19, 2006.
10. M. Dr. S. Vijayarani, "Liver Disease Prediction using SVM and Naïve Bayes Algorithms," *International Journal of Science, Engineering and Technology Research (IJSETR),* vol. 4, no. 4, pp. 1-5, 2015.
11. A. k. D. Pragati Agrawal, "A BRIEF SURVEY ON THE TECHNIQUES USED FOR THE DIAGNOSIS OF DIABETES-MELLITUS," *International Research Journal of Engineering and Technology (IRJET),* vol. 2, no. 3, pp. 1-5, 2015.
12. K. R. S. Sabibullah M, "Prediction of Stroke Risk through Stacked Topology of ANN Model," *International Journal of Advanced Research in Computer Science,* vol. 1, pp. 1-10, 2010.
13. R. A. M. T. F. S. A. M. F. J. K. R. N. N. T. Leila Amini, "Prediction and Control of Stroke by Data Mining," *International journal of preventive medicine ·* , vol. 4, pp. 1-5, May 2013.

14. H. P. E. K. S. P. M. S. P. G. a. A. N. Antonis Lambrou, "Assessment of Stroke Risk Based on Morphological Ultrasound Image Analysis with Conformal Prediction," *IFIP International Federation for Information Processing*, vol. 339, pp. 146-153, 2010.

15. M. o. H. I. R. A. Ohoud Almadani, "Prediction of Stroke using Data Mining Classification Techniques," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 9, no. 1, pp. 1-4, 2018.

16. B. K. a. N. Abdelwahab, "A Model for Predicting Ischemic Stroke Using Data Mining Algorithms," *IJISET - International Journal of Innovative Science, Engineering & Technology*, vol. 2, no. Issue 11, pp. 1-6, November 2015..

17. J. Celko, The Morgan Kaufmann Series in Data Management Systems, 2nd December 2014.

    1. Akiko Aizawa,An information-theoretic perspective of tf–idf measures,Information Processing & Management,Volume 39, Issue 1,2003,Pages 45-65,ISSN 0306-4573,https://doi.org/10.1016/S0306-4573(02)00021-3.
    2. T. Cover and P. Hart, "Nearest neighbor pattern classification," in IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, January 1967. doi: 10.1109/TIT.1967.1053964
    3. Machine Learning by Tom M. Mitchell.
    4. Niwattanakul, Suphakit et al. “Using of Jaccard Coefficient for Keywords Similarity.”
    5. Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. Proceedings of the 23rd International Conference on Machine Learning - ICML ’06. doi:10.1145/1143844.1143874
    6. Goutte C., Gaussier E. (2005) A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In: Losada D.E., Fernández-Luna J.M. (eds) Advances in Information Retrieval. ECIR 2005. Lecture Notes in Computer Science, vol 3408. Springer, Berlin, Heidelberg