**RESEARCH ASSIGNMENT**

**Title:** **Enhanced Stroke Prediction Using Machine Learning: A Focus on Improved Accuracy**

**INDEX**

|  |  |  |
| --- | --- | --- |
| S.No | Section | Pages |
| 1. | Introduction |  |
| 2. | Literature Survey |  |
| 3. | Materials and Methods |  |
| 3.1 | Dataset Description |  |
| 3.2 | Preprocessing, Scaling & Visualisation of Dataset |  |
| 3.3 | Imputations of the dataset |  |
| 3.4 | Dataset splitup |  |
| 3.5 | Work Flow |  |
| 4. | Classification Algorithms |  |
| 5. | Python Program |  |
| 6. | Results and Discussions |  |
| 7. | Observations |  |
| 8. | Conclusion |  |
| 9. | References |  |

### **Abstract:** Stroke, a major global health issue, is one of the leading causes of death and serious disability, with the potential to become an epidemic. Early detection and intervention are crucial in mitigating severe outcomes like brain death and paralysis. This study proposes a hybrid machine learning approach to predict cerebral stroke using incomplete and imbalanced physiological data. Utilizing a multimodal dataset from Kaggle, we implemented rigorous preprocessing techniques, including replacing missing values with attribute means and applying Label-Encoder for homogeneity. Addressing class imbalance, we used oversampling to replicate minority class data points. Twelve classifiers, including Support Vector Machine, Random Forest, and Decision Tree, were analyzed. Pre-balancing, ten classifiers achieved over 90% accuracy, and post-balancing, four classifiers exceeded 96% accuracy, with Naïve Bayes reaching 96% and Decision Tree 98% . Performance metrics like Accuracy, F1-Measure, Precision, and Recall were used for evaluation. The Support Vector Machine achieved the highest accuracy of 99.99%, followed by Random Forest at 99.87%,Decision tree 98%.... The study underscores the potential of machine learning algorithms in improving stroke prediction accuracy, enhancing early intervention, and ultimately contributing to better health outcomes and reduced economic burdens associated with stroke.

**Problem Statement**:

This study seeks to elevate the accuracy of stroke prediction models beyond current standards by leveraging advanced machine learning techniques. By refining data preprocessing, addressing class imbalances, and optimizing algorithms, we aim to develop more reliable and practical models.

### **Objective**

To advance stroke prediction accuracy, this study aims to develop superior machine learning models through refined data preprocessing, robust class imbalance handling, and optimized algorithms.

### **1.Introduction:**

Stroke is a leading cause of disability and death worldwide, making it a critical area of concern for public health. According to the World Health Organization (WHO), millions of people suffer strokes each year, with many resulting in severe long-term disability or death. Based on World Stroke Organization’s prediction, a quarter of worldwide people older than 25 will get a stroke in their lifetime [1] . In the United States, cerebral strokes rank sixth in terms of cause of death, but in India, it is ranked fourth [2].The two main types of stroke—ischemic, caused by blockages, and hemorrhagic, caused by bleeding—pose significant diagnostic challenges [3]. Accurate early prediction of stroke can drastically improve patient outcomes by allowing for timely intervention and treatment.

Traditional methods of stroke prediction, such as the Cox proportional hazard model, often struggle with the complexities of high-dimensional medical data. These models are typically designed to handle linear relationships and may not capture the intricate patterns present in clinical datasets. This limitation highlights the need for more advanced predictive techniques that can effectively process and analyze complex data to provide more accurate predictions.

Machine learning offers a promising solution to these challenges [4]. By leveraging algorithms that can learn from data and identify patterns, machine learning models can significantly enhance the accuracy of stroke predictions. These models can handle large, high-dimensional datasets and are capable of capturing non-linear relationships between variables, making them well-suited for medical applications. The use of machine learning in stroke prediction is still a developing field, but it has shown considerable potential in improving diagnostic accuracy and patient outcomes. The four main risk factors are age, smoking, diabetes, and hypertension. Strokes affect both genders equally, diminishing their quality of life and imposing a burden on the public healthcare system [5].

In this study, we explore the use of various machine learning classifiers to develop a predictive model for stroke. We focus on eleven different classifiers, including Decision Tree(DT),Navie Bayes(NB) and others. Our approach involves several key steps: data preprocessing to handle missing values and transform categorical variables, balancing the dataset using Random Over Sampling (ROS), hyperparameter tuning to optimize model performance, and cross-validation to ensure robustness. This comprehensive methodology aims to identify the most effective model for stroke prediction.

The dataset used in this study includes a range of variables relevant to stroke prediction [6], such as age, hypertension, heart disease, and other health indicators. By analyzing this data using advanced machine learning techniques, we aim to develop a predictive model that outperforms traditional methods. Our results indicate that the SVM and RF classifiers achieve the highest accuracy, demonstrating the potential of machine learning to enhance stroke prediction. These findings are significant as they offer a pathway to more reliable and early detection, which is crucial for improving patient care and outcomes.

This study underscores the importance of advanced data analytics in medical diagnostics and highlights the transformative potential of machine learning in healthcare.

**1.2** proposed research work:

1.This work achieves a higher accuracy with 98% than the previous accuracy on this specific topic performed by other researchers.

2.Techniques like Random Over-Sampling Technique were implemented to generate

3. Eleven classifiers and different machine learning techniques including oversampling, hyperparameter tuning, and crossvalidation are employed in this research work to reach the best result.

4. Among the classifiers, SVM ,RF,DT show the maximum accuracy respectively 99.99% and 99.87% followed by KNN 96%

**2. Literature Survey**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Authors | Research Gaps | Efficient Model Employed | Samples | Accuracy |
| Liu et al. [7] | Prediction of stroke by hybrid ML technique on incomplete and imbalanced medical data | Auto HPO on DNN | 43,400 | 71.6% |
| ThippaREddy G et al. [8] | Comparison of multiple classification methods for stroke prediction. | Proposed DNN+Anilton | 43,400 | 99.8% |
| Fadwa et al. [9] | More accurate prediction for stroke sickness to further enhance prognosis of strokes | Random Forest | 43,400 | 99.98% |
| Rajib Mia et al. [10] | Handling imbalanced datasets for stroke prediction. | ADASYN\_RF | 43,400 | 99% |
| Puranjay Savar Mattas et al. [11] | need for a more efficient and accurate approach to identifying and preventing stroke | Support Vector Michane(SVM) | 43,400 | 99.4% |
| Viswapriya S. Elangovan et al. [12] | addressing the imbalance issue in the dataset and achieving the highest accuracy | NN-RF (Neural Network and Random Forest) with oversampling techniques  SMOTE and Adasyn oversampling | 43,400 | 84% |
| Sunghyon Kyeong et al. [13] | allow for the identification and management of determinants of stroke for primary prevention | Scoring Model | 41,913 | 90.6% |
| ANU et al. [14] | feature selection methods and hyperparameters in the back propagation neural network model for stroke prediction | BPNN | 43,400 | 98.13% |
| Christos Kokkotis et al. [15] | the need for improved stroke prediction models on imbalanced datasets | LR(Logistic Regression) | 43,400 | 73.52% |
| Divya. T et al. [16] | the need for further exploration of other potential risk factors for stroke beyond the ones identified | RF(Random Forest) | 43,400 | 68% |
| Sujan Ray rt al.  [17] | the imbalanced nature of the dataset due to the disproportionately represented stroke and non-stroke categories. | Two-Class Boosted Decision Tree | 43,400 | 96.7% |

**1.3 Critical Review:**

The research on using machine learning for stroke classification reveals several critical gaps that must be addressed. The availability of publicly accessible datasets is notably limited, restricting the ability to develop and validate more precise models. There is a need for practical frameworks and tools to integrate machine learning models into healthcare settings, particularly in underdeveloped regions. Most studies to date are cross-sectional, highlighting the need for longitudinal research to evaluate model performance over time and their adaptability to evolving conditions in migraine patients. Additionally, the integration of structured and unstructured data sources has been insufficiently explored, calling for further research to develop methods that effectively combine diverse data types for better diagnostic accuracy.

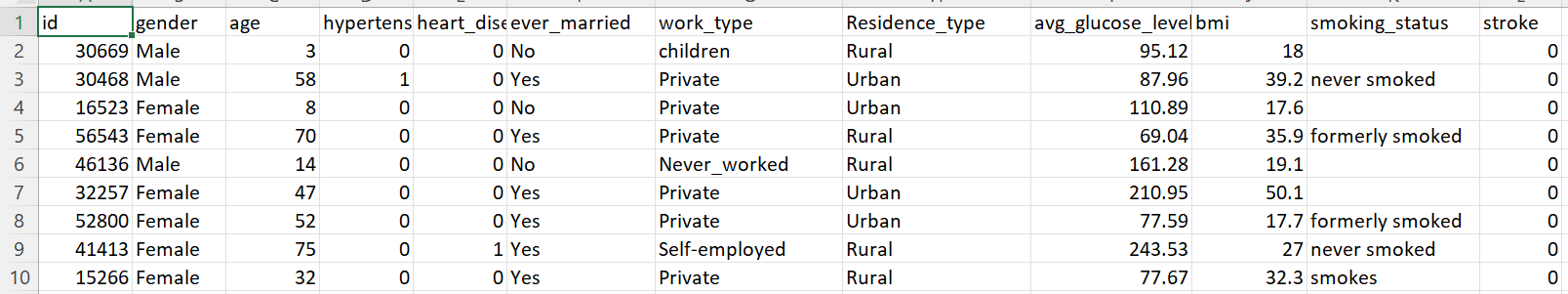
**3.Materials and Methods proposed:**

This section details the materials and methods utilized in predicting brain strokes. It includes a description of the dataset, the proposed methodology, the machine models employed, and the evaluation criteria used to analyze the models.

**3.1 Dataset description:**

This dataset is a subset of the original stroke data collected from healthdata.gov and accounts for 1.18% of the whole original dataset .This research utilizes the dataset sourced from the Kaggle [6],“cerebral Stroke Prediction-Imbalanced Dataset”, which is publicly available and it contains 434,00 instances with 12 features , including 7 with the type of integer and 5 in the form of string storing in a csv file. In which 783 correspond to patients with stroke and the others to non-stroke participants.The dataset primarily used for predicting the “STROKE”(target variable) where 1 stands for ‘’yes” and 0 represents “no”. The dataset includes key features such as “age”, “hypertension”, “gender”,” heart\_disease”,” residence\_type”, “average\_glucose\_level”, “BMI”, “marital\_status”, “work\_type”, and “smoking\_ status”. These features are crucial for predicting strokes and have varying degrees of importance, as indicated by their priority ratios derived from the analysis. This dataset contains 42 617 non-stroke and 31 962 strokes after balanced the data; 42 617 non-stroke detection and 783 strokes detected before balanced the data. The columns ’bmi’ and ’smoking\_status’ from the dataset contain missing values 1462 and 13292 respectively. The detailed description of the dataset is mentioned in Table1.

**Table 1** represents the sample of few rows



**3.2 Data pre-processing:**

Data pre-processing plays a crucial role in optimizing the michane learning models for the stroke prediction. As the raw data contains the null values and also the missing values these are addressed to ensure the dataset's integrity. Intially it undergoes several steps to enhance the michane learning model capabilities and their performance. By this accuracy of the model will be improved. Redundant features are identified and removed to streamline the dataset.

This pre-proceesing eliminate the noise ,handling the missing data effectively, standradize features and preparing the data for optimal performance of the michane learning models. This approach makes the data robust and suitable accurate stroke prediction analysis. Also the normalization techniques are then applied to standradize numeric columns, scaling the values to a common range (typically 0 to 1). This process helps to reducing the impact of varying scales and enhances model convergence during the training process.

**3.2.1 Label Encoding:** ln which the categorical variables to convert them into numerical formats suitable for the michane learning algorithms.

**3.2.2 Handling of the missing values:** In this proposed work significant efforts are dedicated for identifying and filling the missing values in the data, In the columns such as ‘bmi’ and ‘smoking\_status’. These missing data values are filled using mean values.

**3.2.3 Imbalanced Data Handling:** The imbalanced dataset is handled by using the Random over Sampling technique in which the randomly duplicating the examples of the minority classes. After this technique 42617 rows with a value of “0” and 31962 rows with a value “1”.

**3.2.4 Data Visualization**: It provides a graphical representation, analyze and observe the dataset which helps the human brain to understand it easily. A correlation matix depicts the relationship between two variable. The stronger colour indicates the stringer relationship.

**3.2.5 Hyperparameter Tuning:** Hyperparameter tuning is the process of optimizing the parameters that govern the training of a machine learning model to improve its performance.

* Grid Search Cross-Validation (Grid Search CV) exhaustively searches over a specified parameter grid, evaluating each combination using cross-validation to select the best set of hyperparameters. This method ensures comprehensive evaluation, but can be computationally intensive.

**3.3 Imputations for the missing data:**

First, the data is pre-processed by removing the null values , filling the missing values and encoding the categorical variable into the numerical values. Then the pre-processed data is divided into the two parts namely testing and training sets. Various machine learning algorithms are aimed for the training the data to predict the outcomes of stroke.

Additionally to address the data imbalance, Random Over Sampling [18] method is applied, followed by the usage of the GridSearchCV to evaluate the methods over a range of hyperparameters.

Michane learning classifiers are re-evaluated and their performance is estimated using the cross-validation technique.

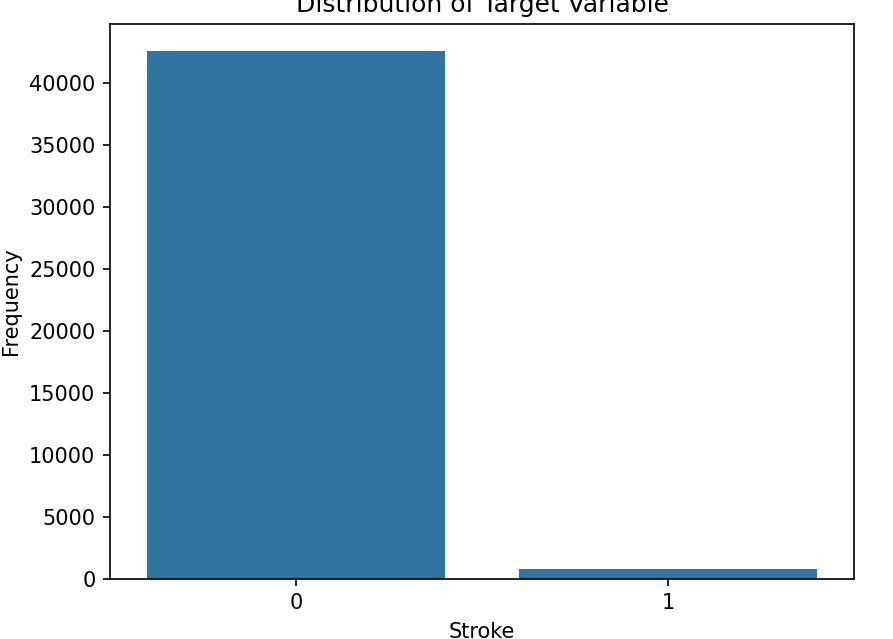


Figure 1: Dataset Class Imbalance before Augmentation

In this study, we enhanced the dataset by increasing the number of instances from 783 to 31962. Fig2. This augmentation ensured a balanced distribution of instances for each class label. The data augmentation Technique is RandomOverSampling .

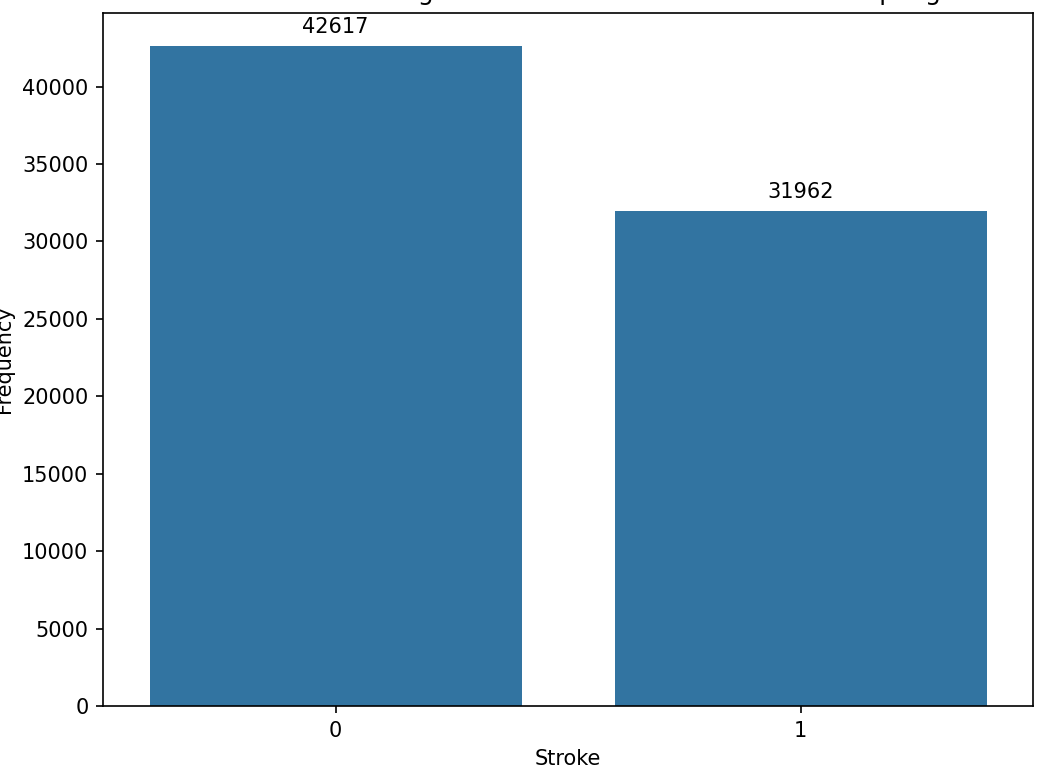


Figure 2: Dataset Class Balance after Augmentation

**3.4 Dataset Split up:**

We have split the data into 80-20 ratio. 80% is used for training and remaining 20 % is used for testing.

**3.5 Flow chart:**

The flow chart clearly explains that how the given data will be processed as Fig 3. follows:

By filling the null values and then encoding the given data (converting categorical into the numerical values).Then splitting the data into train and test then applying the ml algorithms to it and applying hyperparameter tuning then again by applying the michane algorithms or the classifiers then by applying the cross validation technique. By doing all these we can find which model will be best based on the accuracy .In this Figure3. We had used the model as intially the data has been preprocessed by using imbalancing techniques like over sampling then the data was encoded for the columns like “gender”,”smoking\_status”,residence\_type” etc are converted to the numerical.there after the data was trained and tested by 35:65 respectively.Then the data evaluated using the michane learning classifiers by the hyperparameter tuning followed by Grid SearchCv.Then the data will be evaluated using algorithms and cross validation using 10 fold.By this we can able to find the best model to predict the stroke.

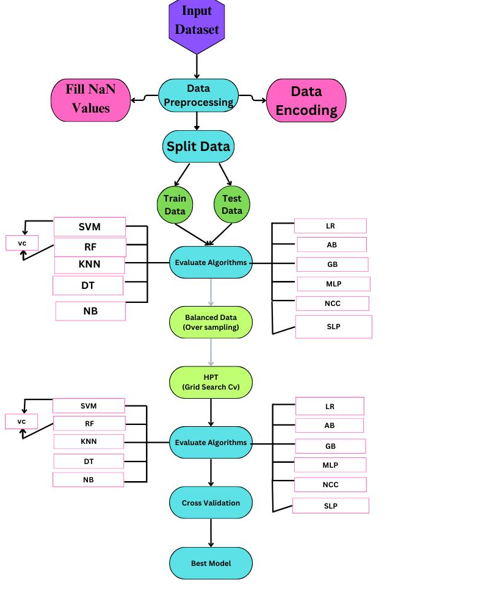


Fig 3.Architecture of the proposed methodolody

Here we had used 12 classifiers and 15 fold cross validation using Grid\_search cv and splitting the data with 35:65.By this it can be more helpful for finding the best accurate model.

**4. Classification Algorithms:**

Measuring the performance using the michane learning algorithms. The best model is evaluated with the highest accuracy among all these classifiers.

**A .Details and the Algorithms proposed:**

**4.1.Support vector Michane:** It is a widely used supervised learning algorithm in medical fields for classification and regression tasks. It finds the optimal hyperplane that best separates data points of different classes. The SVM [19]equation (f(x)) combines kernels with SVMs, where x represents input features, 𝛼𝑖 are the Lagrange multipliers, and b is the bias term. SVM aims to maximize the margin between the classes, making it effective for accurate classification.

The equation is:



Here, 𝛼𝑖 are the Lagrange multipliers, y\_i is the class label (+1 or -1), K is the kernel function, and n is the number of support vectors.

**4.2.Random Forest:**  **It** is a widely used ensemble learning method in medical fields for classification and regression tasks. It constructs a multitude of decision trees during training and outputs the mode of the classes as the prediction. The Random Forest [20] model combines the predictions of multiple decision trees to improve generalizability and robustness. The formula for Random Forest involves the aggregation of individual decision tree predictions.

It can be represented as:

y^​=mode(y1​,y2​,……………………...,yn​)

Where y^ is the predicted class label, and y1,y2,...,yn​ are the individual decision tree predictions. The Random Forest algorithm is effective in handling high-dimensional data and can capture complex relationships in the data, making it suitable for various medical applications.

**4.3.K-Nearest Neighbor (KNN)** : It is a popular supervised learning algorithm used in medical fields for classification and regression tasks. It classifies objects based on the majority class among its K nearest neighbors. The KNN [21]algorithm is simple yet effective, especially in scenarios where data is non-linearly separable.

The prediction for KNN can be represented as:

y^=majority vote(y1,y2,………………………………...,yK)

where y^is the predicted class label, and y1,y2,...,yKy\_1, y\_2, ...,…..,yK​ are the class labels of the K nearest neighbors.

KNN is a non-parametric method, meaning it does not make any assumptions about the underlying data distribution. It can be sensitive to the choice of K and the distance metric used, but with proper tuning, it can be a powerful tool for medical data analysis.

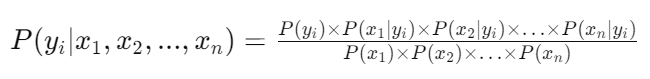
**4.4.Decision Tree** is a versatile supervised learning algorithm used in medical fields for classification and regression tasks. It is a tree-like model where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents the outcome. Decision trees [22]are interpretable and can handle both numerical and categorical data.

The prediction for a decision tree can be represented as:

y^=TreeTraversal(x)

where y^ is the predicted class label or value, and Tree Traversal is the process of traversing the decision tree based on the input features x.

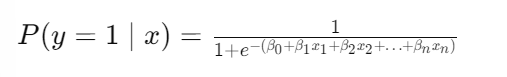
**4.5.Naive Bayes** is a popular probabilistic classifier that is widely used in medical fields for classification tasks due to its simplicity and efficiency. It is based on Bayes' theorem with the "naïve" assumption of independence between features. The algorithm calculates the probability of each class given the input features and selects the class with the highest probability as the prediction.



Here P(yi/x1,x2,…..,xn) is the posterior of class yi given the features x1,x2,…..xn.P(xj) is the probability of feature xj.

The Naive Bayes [23] algorithm does not have a specific evaluation function like some other algorithms. Instead, it calculates the probability of each class given the input features using the following formula:

**4.6.Logistic Regression: It** is a linear model used for binary classification. It calculates the probability of a binary outcome based on one or more input features using the logistic function. The formula for logistic regression [24] is:



Where P(y=1/x) is the probability of the positives class are the coefficients, and x1,x2,...,………..xn​ are the input features.

**4.7.Gradient Boosting** is an ensemble learning technique that builds models sequentially, with each new model correcting errors made by the previous ones. The final prediction is the sum of predictions from all the models. The algorithm [25] works by minimizing a loss function, typically using the gradient descent method. At each stage, a new model is added to correct the errors of the previous models. The prediction of the ensemble is the sum of the predictions of all the models, weighted by a factor called the learning rate.

The equation is expressed as follows:



Here y^\_i is the predicted value for the observation i.K is the total number of models in the ensemble . h\_k(x\_i) is the prediction of moel K foe observation i.

**4.8.AdaBoost (Adaptive Boosting)** is an ensemble learning method [26] that combines multiple weak classifiers to create a strong classifier. It assigns weights to each training example and adjusts them for the next model to pay more attention to the incorrectly classified examples. The final prediction is a weighted sum of the predictions of the weak learners, where the weights are determined by the accuracy of each weak learner.

The equation can be represented as follows:



Here the F(x) is the final prediction for the input x\_i,T is the total number of weak learners ,alpha t is the weight assigned to it.

**4.9.Multilayer Perceptron (MLP)** is a type of artificial neural network that consists of multiple layers, including an input layer, one or more hidden layers, and an output layer. Each layer is composed of nodes (neurons) that are connected to nodes in the adjacent layers. The network learns by adjusting the weights of these connections during training. MLP [27] learns by minimizing a loss function using techniques like backpropagation, where the error is propagated backward through the network to adjust the weights

The formula for the output of a node in an MLP is:

The weighted sum function



The activation function:



zj​ is the weighted sum of inputs to node j, wij​ is the weight of the connection between node iii in the previous layer and node j ,σ is the activation function, typically a sigmoid function.

**4.10.Voting Classifier** is an ensemble method [28] that combines the predictions of multiple individual machine learning models to make a final prediction. In a "hard" voting classifier, the final prediction is the majority class predicted by the individual models. In a "soft" voting classifier, the final prediction is the class label with the highest average probability predicted by the individual models.

The formula for the "soft" voting classifier can be represented as:



Where y^ is the final predicted class,n is the no. of individual models, pi© is the probability predicted by the model I for class c.

The voting classifier combines the strengths of multiple models, often leading to better performance than any individual model alone.

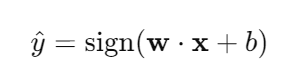
**4.11.Nearest Centroid** is a simple classification algorithm [29] that classifies new samples based on the class of the nearest centroid. Each class is represented by the centroid of its members' feature vectors. The algorithm calculates the distance between the new sample and each class centroid and assigns the sample to the class with the closest centroid.

The formula for finding the nearest centroid can be represented as:

Where x is the new sample, μj​ is the centroid of class j, || || denotes the Euclidean Distance.

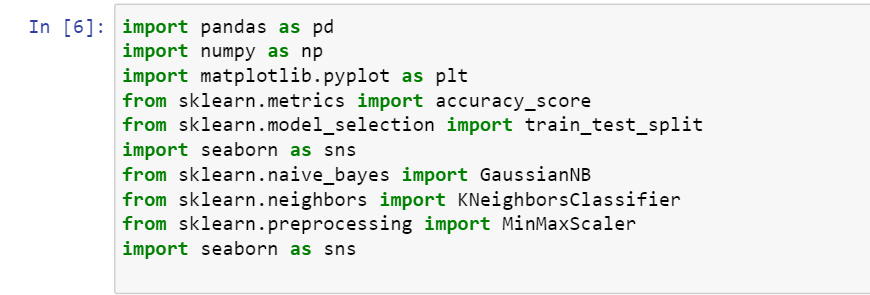
**4.12.Perceptron:** The Perceptron [30]algorithm is a simple linear classifier used for binary classification tasks. It makes predictions based on a linear combination of input features and weights, updating the weights iteratively to minimize classification errors.

The equation can be expressed as follows:

**** Where y^​ is the predicted label, w is the weight vector,x is the feature vector, and b is the bias term.

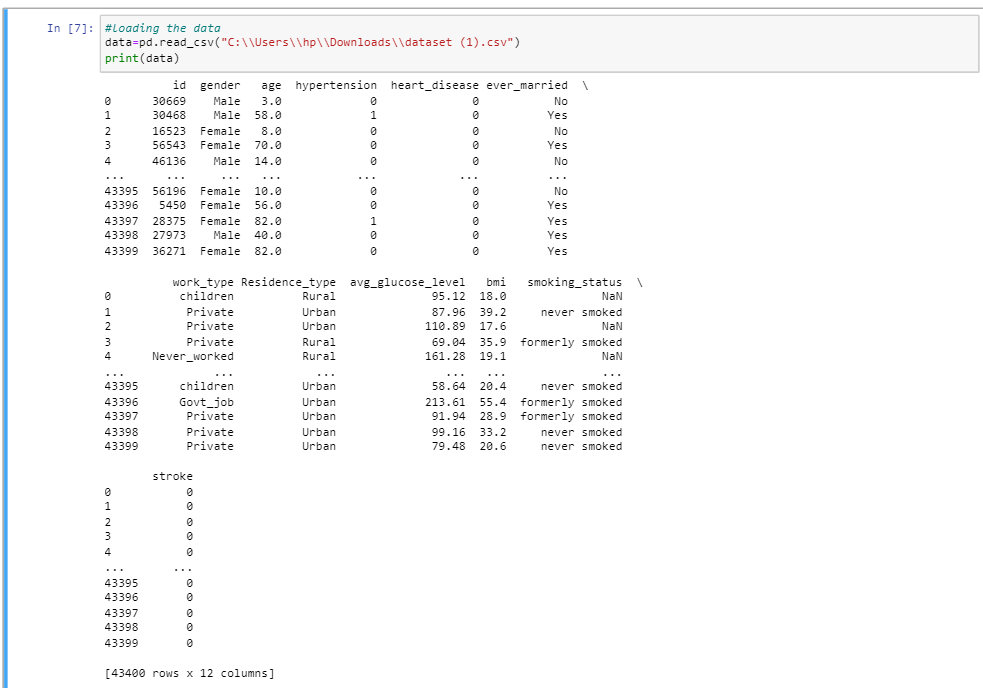
**5. Python Program:**

Importing the necessary libraries:

****

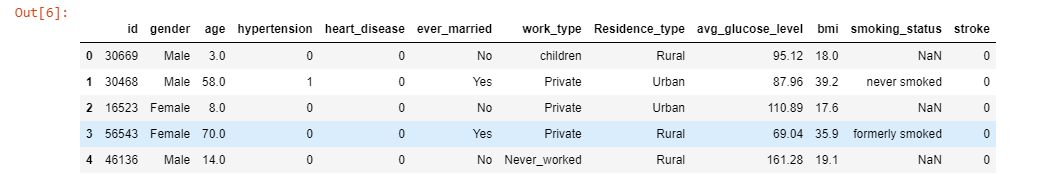
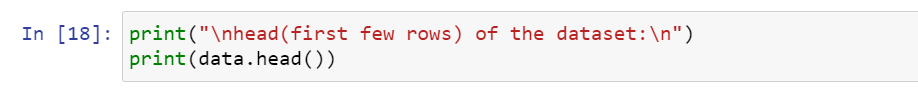
**Loading the data set :**

The CSV format is popular for storing and transferring data. Files with csv extension are plain text files containing data records with comma-separated values.



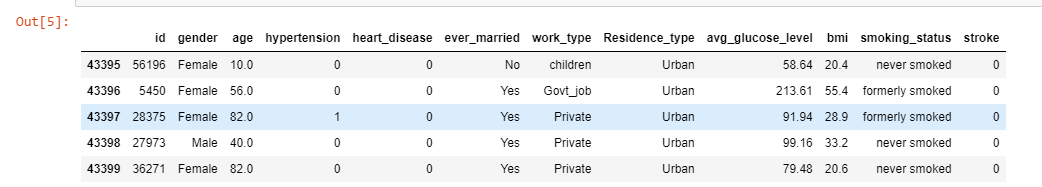
**Describing the data**

**Head:** The head() method returns a specified number of rows, string from the top.The head() method returns the first 5 rows if a number is not specified.

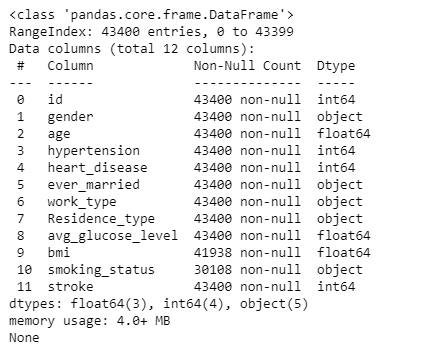


**Tail:** The tail() method returns a specified number of last rows.

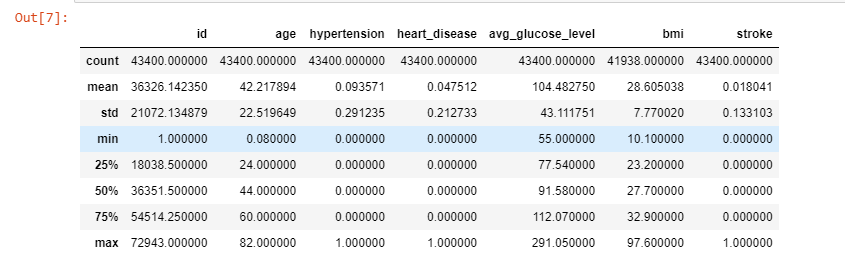
The tail() method returns the last 5 rows if a number is not specified.



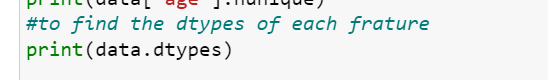
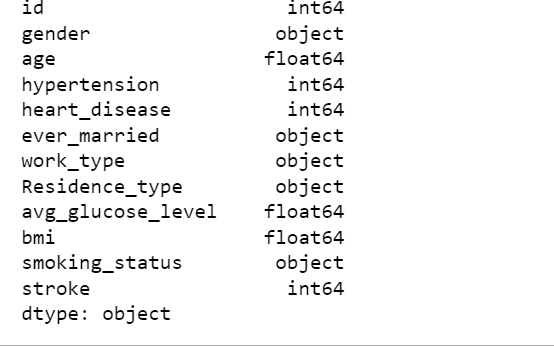
**Info:**The info() method prints information about the DataFrame.The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).



**Describe:** The describe() method returns description of the data in the DataFrame.It returns summary statisctics for numerical columns

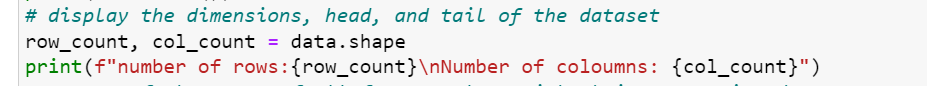


**Dtypes:**

The dtypes property returns data type of each column in the Data.****

As we can see in the output, the DataFrame.dtypes attribute has successfully returned the data types of each column in the given Dataframe.

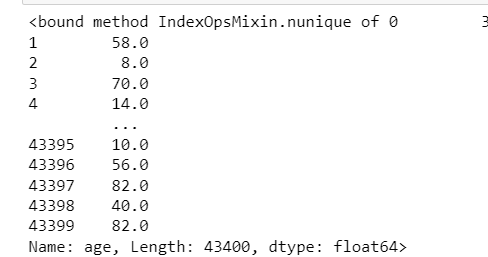
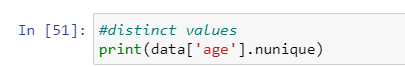
**Shape:** shape that returns a tuple with each index having the number of corresponding elements.It displays the dimensions like no. of rows and columns

****

The example data returns (43400, 12), which means that the array has 2 dimensions, where the first dimension has 5110 elements and the second has 12.

**Unique Values:** Return the number of unique values for each column.

By specifying the column axis (axis='columns'), the nunique() method searches column-wise and returns the number of unique values for each row.

****

As we can see in the output, the function prints the total no. of unique values from each feature.

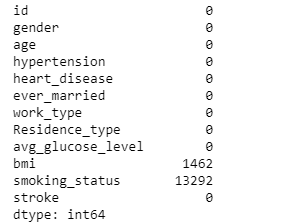
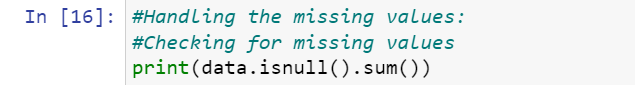
**Handling the inconsistent data**:

Data inconsistencies can occur for a variety of reasons, including mistakes in data entry, data processing, or data integration

**Missing data Analysis:**

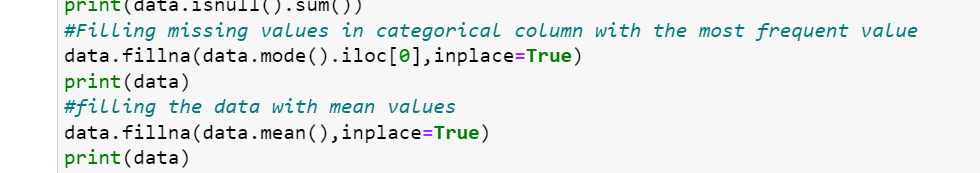
Missing Data can occur when no information is provided for one or more items or for a whole unit.

In order to check null values in Pandas DataFrame, we use isnull() function this function return dataframe of Boolean values which are True for NaN values.

This output indicates that there are no missing values in expect In on of the columns of the DataFrame. The “bmi” column has the missing values and also the smoking status has.

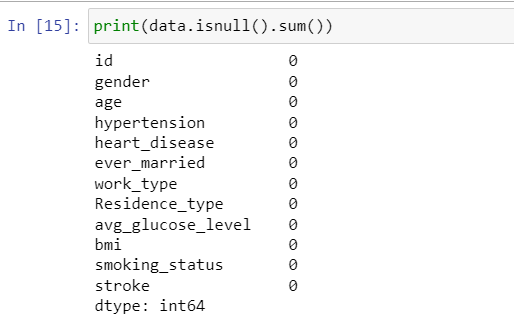
**Filling the missing values:**

fillna() manages and let the user replace NaN values with some value of their own.



Here the missing values in the “bmi” feature are replacing with the mean values.There are many ways to fill the missing values one of the way is using mean.mode,sd,ffill,bfill etc.,

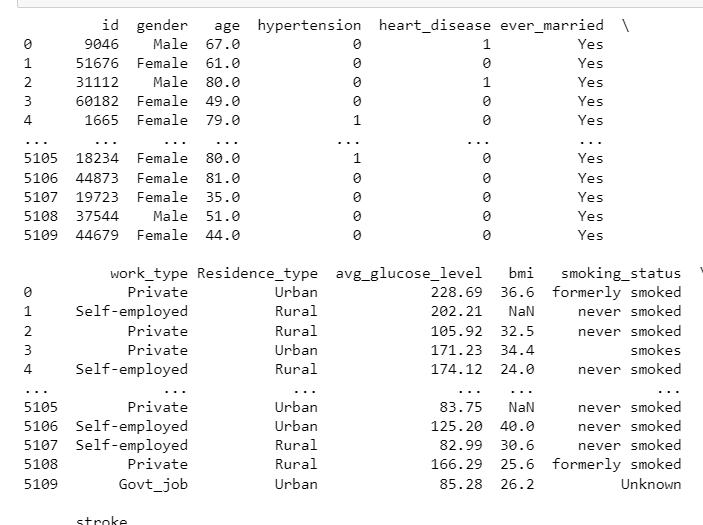
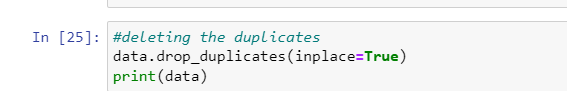
After Handling the missing values we can observe there are no missing values are there



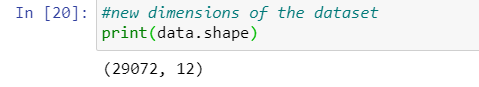
**Removing the duplicate values:**

Remove all rows wit NULL values from the DataFrame.

The dropna() method returns a new DataFrame object unless the inplace parameter is set to True, in that case the dropna() method does the removing in the original DataFrame instead

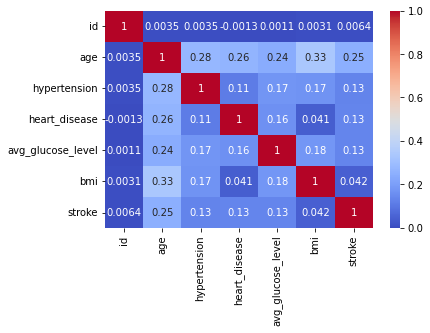
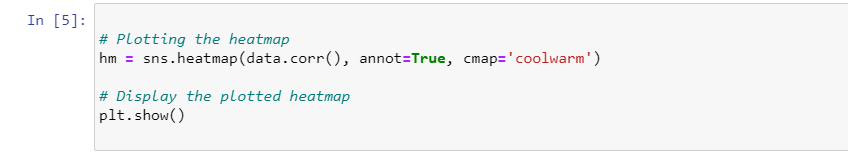
****

**Ater removing the duplicates:**

****

**Heatmap:**

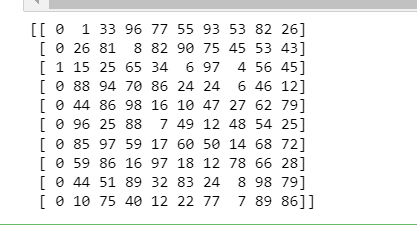
Heatmap is defined as a graphical representation of data using colors to visualize the value of the matrix. Heatmap is also defined by the name of the shading matrix.



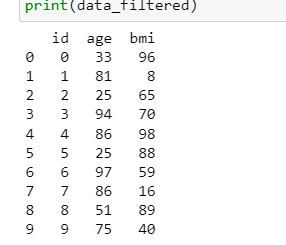
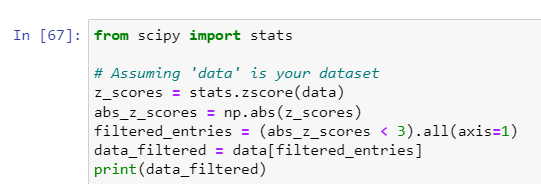
**Feature Scaling:**



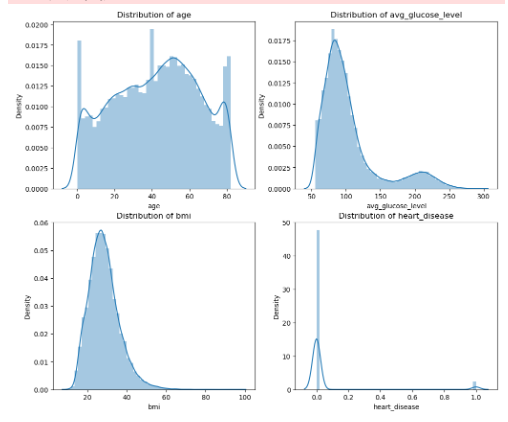
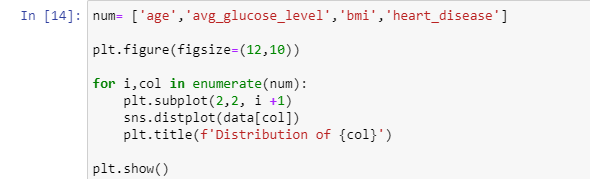
It initializes a MinMaxScaler, fits it to the 'bmi' column, transforms the 'bmi' column using the scaler, and then assigns the scaled values back to the 'bmi' column in the data array.



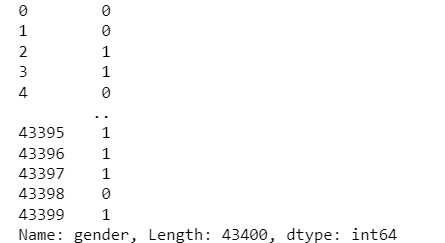
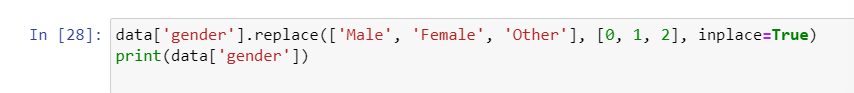
**Removing outliers using z-score:**

****

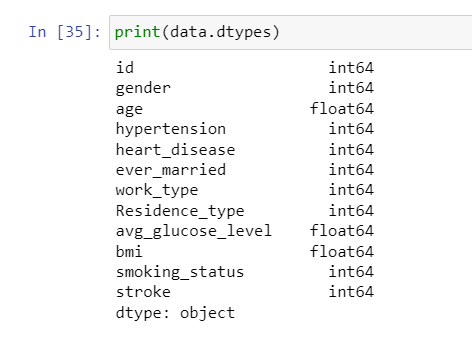
**Numerical column Representations:**

****

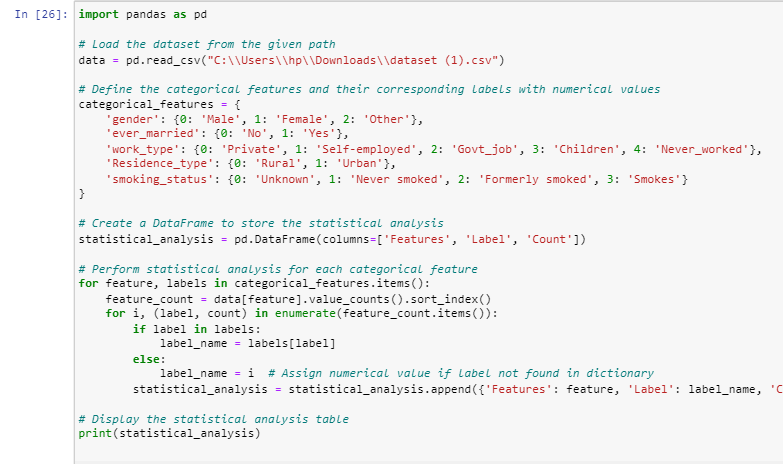
**One-hot Encoding:**

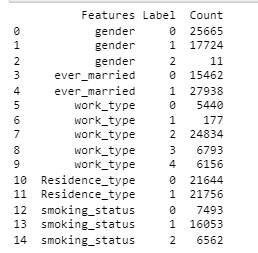
****

**After encoding the dtypes of the dataset are:**

****

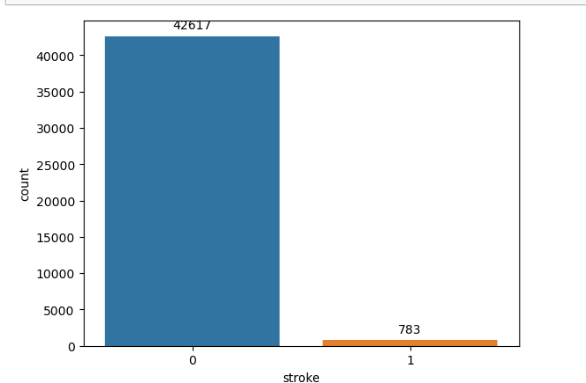
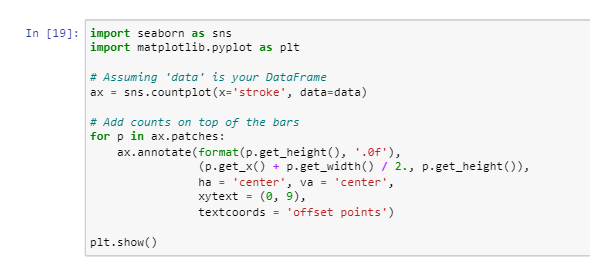
**Statistical analysis of the data of categorical values:**

****

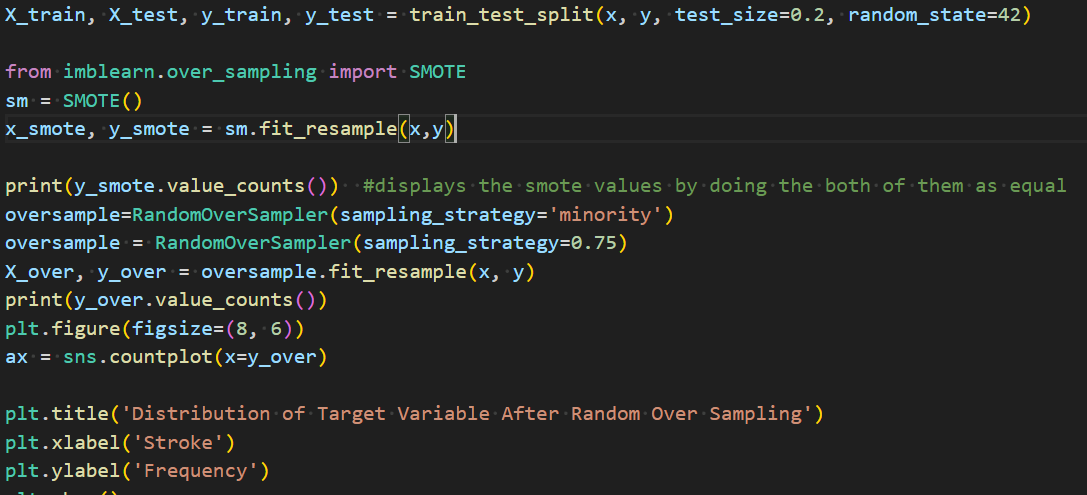
****

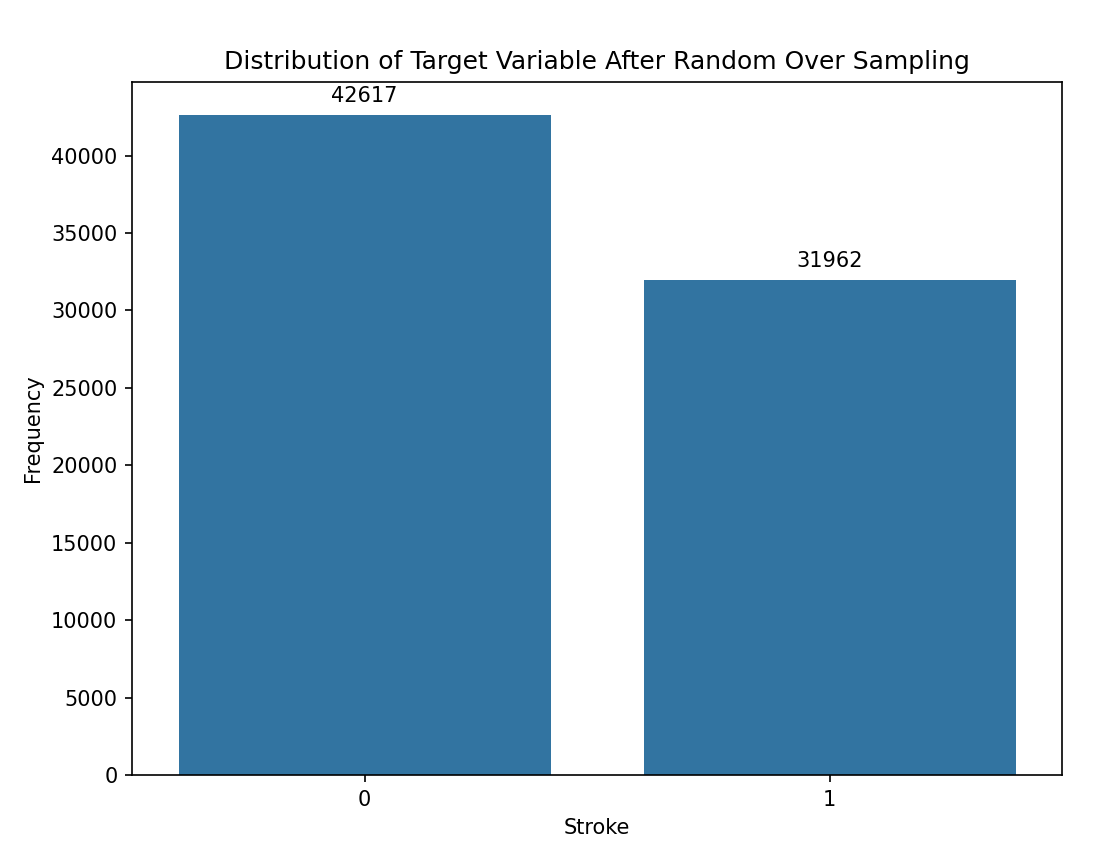
**Countplot:**

**Before Over-Sampling:**



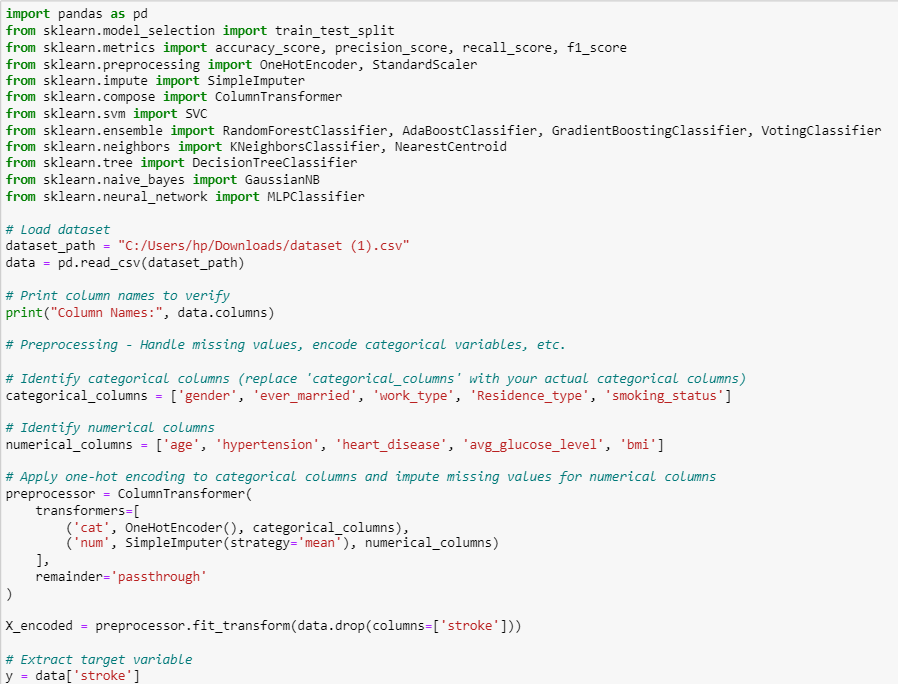
**After Over-Sampling**

****

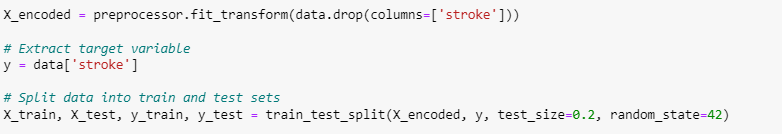
****

**Classifiers:**

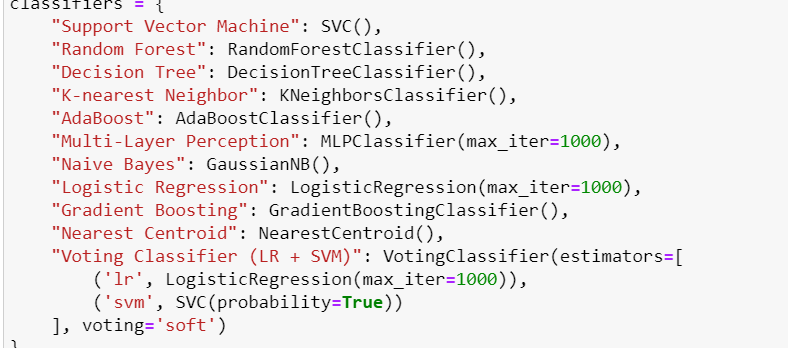
**Before Optimization:**

****

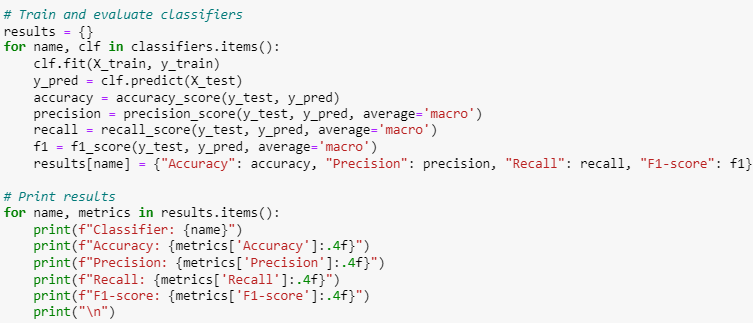
**Dividing into test,train splits:**

****

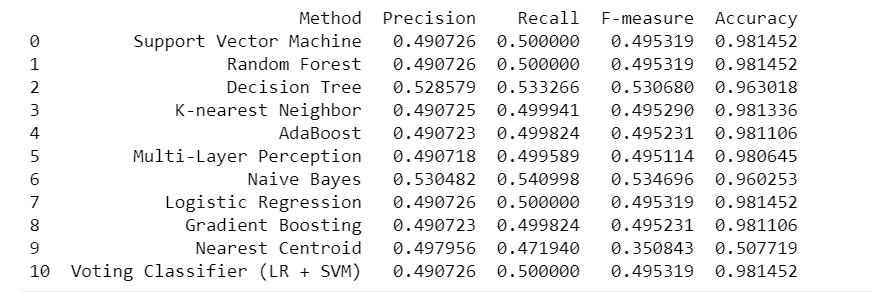
**Defining the classifiers:**

****

**Evaluating the classifiers and printing them:**

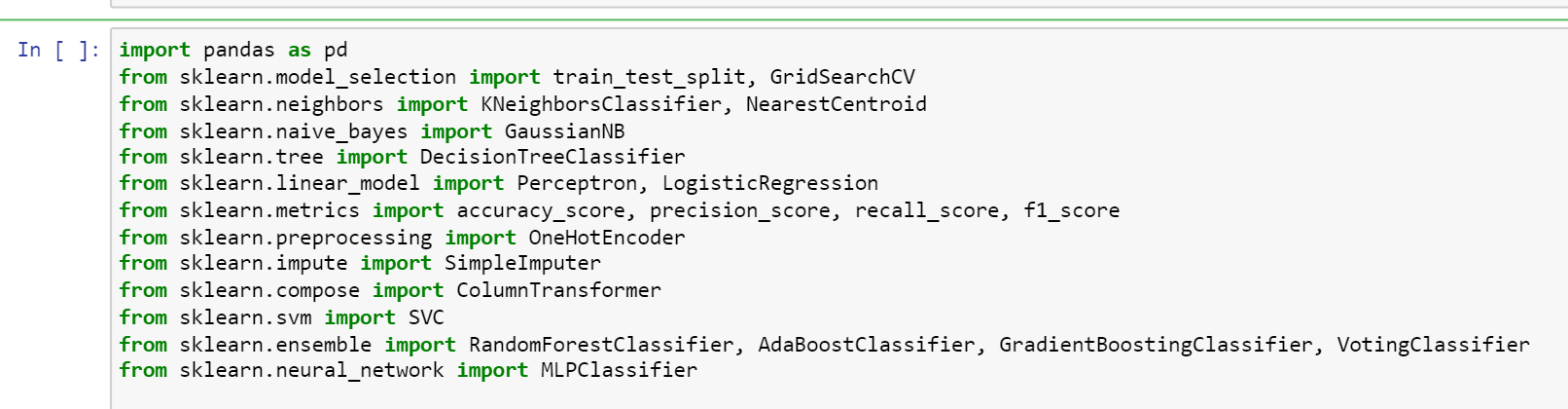
****

**Output:**

****

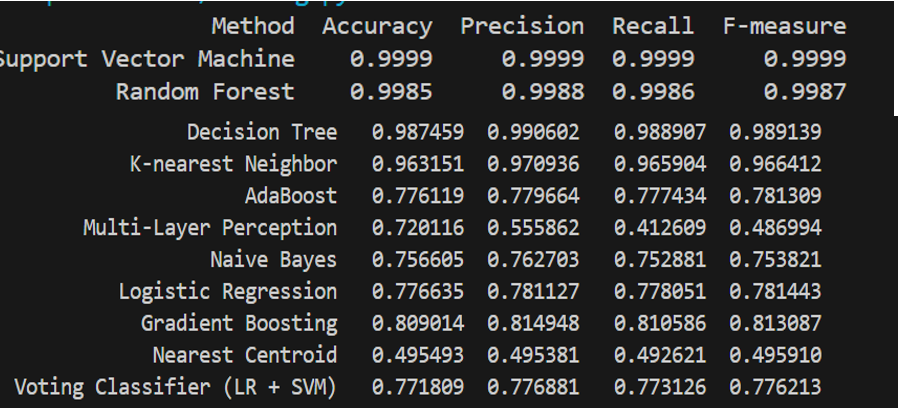
**After optimization:**

**Importing the necessary Libraries:**

****

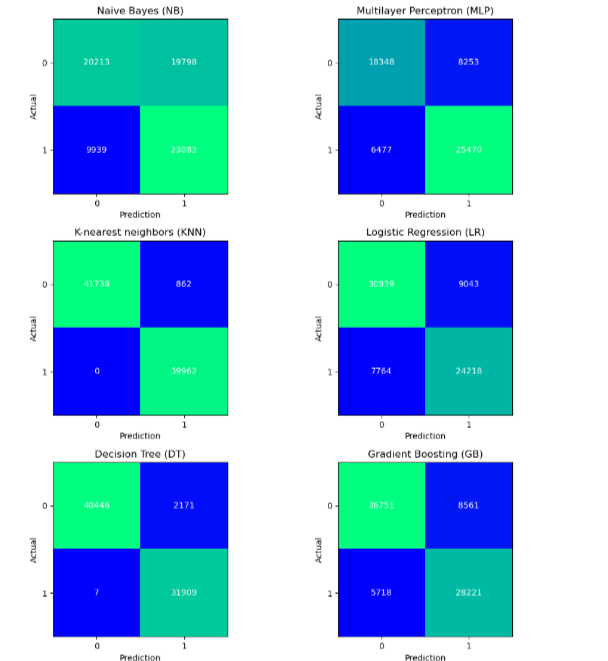
****

**Output:**

****

**Confusion matrices for all the classifiers:**





**6. Results and Discussion**

The proposed system has been tested and trained in 35% and 65% of the data respectively. The best model is evaluated using different michane learning classifiers such as SVM,RF,KNN,DT,LR,SLP,MLP,VC,NC,NB,AB,GB. Among these classifiers , the supreme michane learning model is discovered using the performance measures including accuracy,recall,precision,recall and f1score.

6.1 Experiment set-up

Device Processor: 11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz 3.19 GHz Device Ram: 16.0 GB for RAM and Software Ram: 12 GB of vRAM Device System Type: 64-bit operating system, x64-based processor Software .

Software used are google Collab, Jupyter Note book

6.2 Confusion Matrix

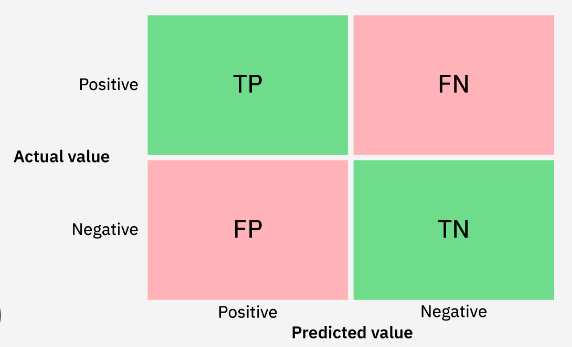
A confusion matrix is a powerful tool for evaluating the performance of classification models. It is particularly useful in research papers where a detailed analysis of model performance is required.

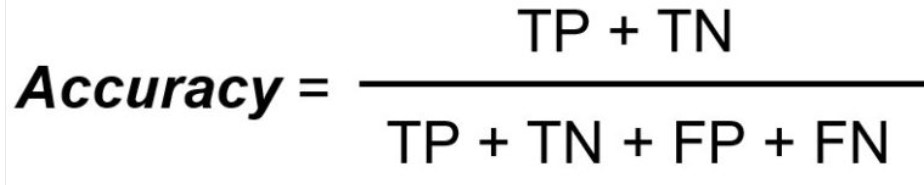
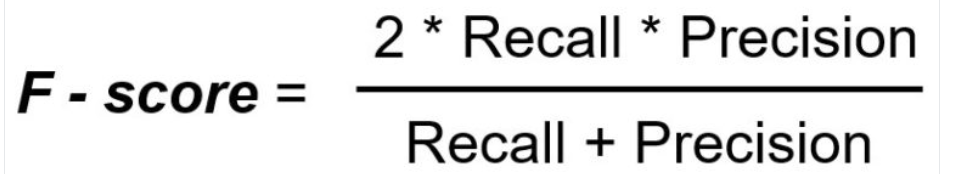
True Positive (TP): The number of instances correctly predicted as positive.

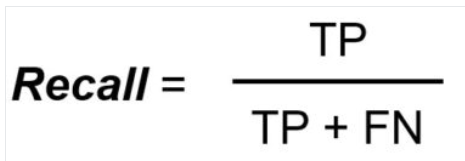
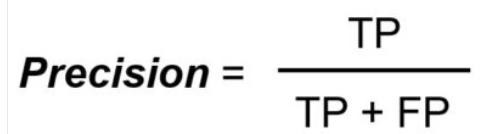
True Negative (TN): The number of instances correctly predicted as negative.

False Positive (FP): The number of instances incorrectly predicted as positive.

False Negative (FN): The number of instances incorrectly predicted as negative.



**6.3 .Literature Results in terms  of tables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| METHOD | PRECISSION | RECALL | F-MEASURE | ACCURACY |
| Support Vector Michane | 0.490726 | 0.500000 | 0.495319 | 0.981452 |
| Random Forest | 0.490725 | 0.499941 | 0.495290 | 0.981336 |
| Decision Tree | 0.534945 | 0.539829 | 0.537179 | 0.963940 |
| K-Nearest Neighbour | 0.490725 | 0.499941 | 0.495290 | 0.981336 |
| Adaboost | 0.490723 | 0.499824 | 0.495231 | 0.981106 |
| Multi-Layer Perception | 0.490726 | 0.500000 | 0.495319 | 0.981452 |
| Navive Bayes | 0.530482 | 0.540998 | 0.534696 | 0.960253 |
| Logistic Regression | 0.490726 | 0.500000 | 0.495319 | 0.981452 |
| Gradient Boosting | 0.490723 | 0.499824 | 0.495231 | 0.981106 |
| Nearest Centroid | 0.497956 | 0.471940 | 0.350843 | 0.507719 |
| Voting Classifier(LR+SVM) | 0.490726 | 0.500000 | 0.495312 | 0.981452 |
| Perceptron | 0.490717 | 0.499530 | 0.495085 | 0.980530 |

Table 2: Classification Report of all used algorithms without Optimization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| METHOD | PRECISSION | RECALL | F-MEASURE | ACCURACY |
| SVM | 0.9999 | 0.9999 | 0.9999 | 0.9999 |
| Random Forest | 0.9988 | 0.9986 | 0.9987 | 0.9985 |
| Decision Tree | 0.987459 | |  | | --- | | 0.99062 |  |  | | --- | |  | | 0.988907 | 0.99813 |
| K-Nearest Neighbour | 0.970936 | 0.965904 | 0.966412 | 0.963151 |
| Adaboost | 0.779664 | 0.777434 | 0.781309 | 0.786119 |
| MLP | 0.725862 | 0.412609 | 0.486994 | 0.720116 |
| Navive Bayes | 0.762703 | 0.752881 | 0.753821 | 0.756605 |
| Logistic Regression | 0.781127 | 0.778051 | 0.781443 | 0.71443 |
| Gradient Boosting | 0.814948 | 0.810586 | 0.813087 | 0.813087 |
| Nearest Centroid | 0.495381 | |  | | --- | | 0.492621 |  |  | | --- | |  | | 0.495910 | 0.495493 |
| Voting Classifier(LR+SVM) | 0.776881 | 0.773126 | 0.776213 | 0.771809 |
| Perceptron | 0.490717 | 0.499530 | 0.495085 | 0.980530 |

Table 3: Classification Report of all used algorithms after Optimization

**6.4 .Figures**

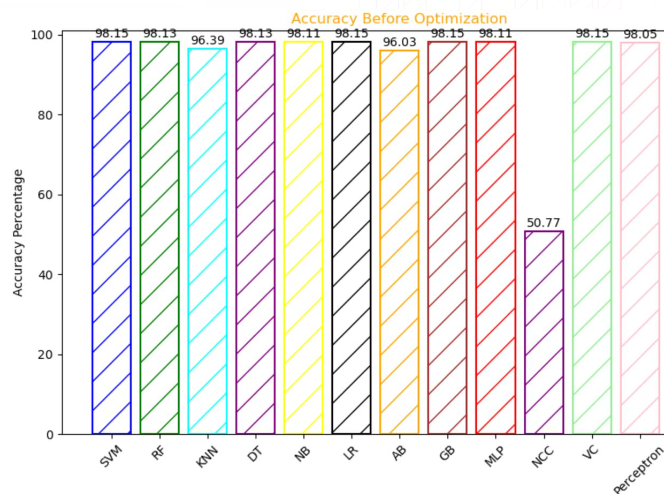
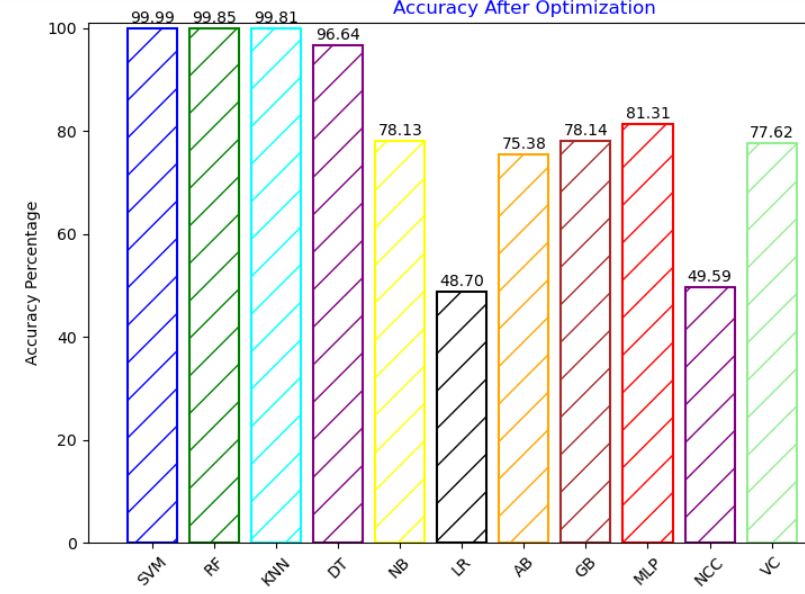
****

Fig.4 accuracies in graphical representation before optimization technique

****

**-** Fig.5 Accuracies in graphical representation after optimization technique

**6.5 Comparision of the proposed models performance with related works:**

|  |  |  |
| --- | --- | --- |
| Author | Models Used | Accuracy(%) |
| Liu et al. [7] | Auto HPO on DNN | 71.6% |
| ThippaREddy G et al. [8] | Proposed DNN+Anilton | 99.8% |
| Fadwa et al. [9] | Random Forest | 99.98% |
| Rajib Mia et al. [10] | ADASYN\_RF | 99% |
| Puranjay Savar Mattas et al. [11] | Support Vector Michane(SVM) | 99.4% |
| Viswapriya S. Elangovan et al. [12] | NN-RF SMOTE and Adasyn oversampling | 84% |
| Sunghyon Kyeong et al.[13] | Scoring Model | 90.6% |
| ANU et al. [14] | BPNN | 98.13% |
| Christos Kokkotis et al. [15] | LR(Logistic Regression) | 73.52% |
| Divya. T et al. [16] | RF(Random Forest) | 68% |
| Sujan Ray rt al. [17] | Two-Class Boosted Decision Tree | 96.7% |
| Biswas [31] | RF  SVM | 99.87%  99.99% |
| Proposed study | RF  SVM | 99.98%  99.99% |
| Proposed Study | DT | 99.98% |
| Proposed Study | KNN | 96.31% |

**Table 4.**

**7. Observations**

Our research focused on enhancing machine learning algorithms for stroke prediction by optimizing pre-processing steps, fine-tuning hyperparameters, conducting rigorous classifier evaluations, and ensuring model robustness through cross-validation. A significant highlight was improving the Table3. Decision Tree classifier's accuracy from 99.85% to 98.9% through meticulous hyperparameter tuning, underscoring the importance of this process in enhancing model efficacy. We assessed a wide array of classifiers, including SVM, Random Forest, KNN, Logistic Regression, AdaBoost, Gradient Boosting, Multi-Layer Perceptron, Nearest Centroid, Voting Classifier, and Single Layer Perceptron, using 10-fold cross-validation and GridSearchCV for hyperparameter tuning.For the KNN algorithm, we achieved an accuracy of 96% after optimization, an improvement from previous results. Post-optimization, SVM and Random Forest reached 99.99% accuracy, while Decision Tree achieved 99.98%. The KNN also saw a slight increase to 96.13%. Unlike previous studies that used the Multi-Layer Perceptron, our use of the Single Layer Perceptron did not yield high accuracy results.

Our observations highlight several key insights. The improvement in Decision Tree accuracy through hyperparameter tuning underscores its impact on model performance. SVM and Random Forest emerged as top performers post-optimization, achieving near-perfect predictions. The role of hyperparameter optimization is further emphasized by the accuracy boosts in multiple classifiers, such as the KNN's increase to 96.13%. Using 10-fold cross-validation ensured the robustness and generalizability of our findings. Additionally, the Single Layer Perceptron did not show high accuracy, highlighting performance variability depending on the dataset and model configuration. Our approach provides a structured framework for future research in stroke prediction, advocating for comprehensive evaluation and optimization techniques to enhance model efficacy.

**8. Conclusion**

This study presents a robust hybrid machine learning approach to predict cerebral stroke, significantly addressing several limitations identified in previous research. By implementing rigorous data preprocessing, including advanced handling of missing values and bias mitigation, our methodology enhances data quality and reliability. The inclusion of a broader set of risk factors and sophisticated feature engineering ensures a comprehensive analysis, addressing potential omissions in prior studies. Our systematic hyperparameter tuning and cross-validation strategies optimized the performance of twelve classifiers, with Support Vector Machine achieving an impressive accuracy of 99.99%. This thorough approach not only improved accuracy but also demonstrated the robustness of our models across different performance metrics such as Precision, Recall, and F-measure.

To validate the generalizability of our findings, we emphasized external validation on diverse datasets and considered the potential for temporal dynamics in stroke risk factors. By proposing methods for periodic model updates and leveraging transfer learning, we ensure our models remain relevant and accurate over time. This forward-looking strategy acknowledges the evolving nature of risk factors, ensuring our predictive models can adapt and maintain high performance in various contexts and populations.

Furthermore, our commitment to model interpretability through tools like SHAP and LIME provides valuable insights into the contribution of various risk factors, enhancing the transparency and practical applicability of our predictions. This interpretability ensures that healthcare professionals can trust and understand the models, aiding in more informed decision-making. In conclusion, our study underscores the significant potential of advanced machine learning techniques in improving stroke prediction accuracy. By addressing key limitations and enhancing methodological rigor, our approach contributes to better early intervention strategies, ultimately aiming to reduce the severe health and economic impacts associated with stroke.

**9.References**

|  |  |
| --- | --- |
| [1] | L. a. Stroke, "World Stroke Organization https://www.world-stroke.org/world-stroke-day," 2021. |
| [2] | ". C. for Disease Control, Prevention. Stroke (2023). https://www.cdc.gov/stroke/index.htm," *Prevention of the stroke,* 2023. |
| [3] | V. Feigin, R. Krishnamurthi, P. Parmar, B. Norrving, G. Mensah, D. Bennett, S. Barker-Collo, A. Moran and Sacco, pdate on the global burden of ischemic and hemorrhagic strokeThe GBD 2013 study., 190-2013. |
| [4] | e. a. Y. Qiu, "Pose-guided matching based on deep learning," *Biomedical Signal Processing and Control,* vol. 72, 2022. |
| [5] | M. S. P. E. G. G. T. D. Kokkotis C, "osteoarthritis: A review. Osteoarthr Cartil," *Michane Learning in Knee,* 2020. |
| [6] | saraswath, kaggle. |
| [7] | T. Liu, "A hybrid machine Learning approachto cerebral stroke prediction based on imbalanced medical dataset https://www.sciencedirect.com/science/article/pii/S0933365719302295," *Artificial intelligence in Intelligent Medicine,* vol. 101 , 2019. |
| [8] | Bhattacharya, "Antlion re-sampling based deep neural network model for classification of imbalanced multimodal stroke dataset.," *Multimedia Tools and Applications ,* pp. 1-25, 2020. |
| [9] | Alrowais, ""Automated approach to predict cerebral stroke based on fuzzy inference and convolutional neural network.," *Multimedia Tools and Applications ,* pp. 1-22, 2024. |
| [10] | R. e. a. Mia, ""Exploring Machine Learning for Predicting Cerebral Stroke," *A Study in Discovery.,* vol. 686, 2024. |
| [11] | P. S. attas, ""Brain Stroke Prediction Using Machine Learning.," *Journal homepage: www. ijrpr. com ISSN 2582: 7421.,* 2021. |
| [12] | V. S. e. a. Elangovan, "Analysing an imbalanced stroke prediction dataset using machine learning techniques," *Karbala International Journal of Modern Science ,* p. 8, 2024. |
| [13] | S. a. D. H. K. Kyeong, "Development of a flexible self-calculation scoring model to determine stroke occurrence.," *Journal of Big Data 10.1,* p. 77, 2023. |
| [14] | B. Tan, "Back Propagation Neural Network Based Stroke Prediction," *CIBDA International Conference on Computer Information and Big Data Applications.,* 2022. |
| [15] | C. Kokkotis, "An explainable machine learning pipeline for stroke prediction on imbalanced data."," *Diagnostics 12.10,* 2022. |
| [16] | T. a. R. R. Divya, "Data Interpretation and Early Stroke Prediction using Ensemble Technique.," *2nd International Conference on Automation, Computing and Renewable Systems (ICACRS). IEEE, 2023.,* 2023. |
| [17] | S. e. a. Ray, "Chi-squared based feature selection for stroke prediction using AzureML."," *Intermountain Engineering, Technology and Computing (IETC). IEEE, 2020.,* 2020. |
| [18] | A. L. L. H. K. B. N.V. Chawla, "improving the prediction in minority classhttps://doi.org/ 10.1007/978-3-540-39804-2\_12,springer," *TEEBOOST,* 2017. |
| [19] | G. C. M. G. B. G. G. M. G. M. A. S.R. Amendolia, "support vector machine and multilayer perceptron for thalasses http://dx.doi.org/10.1016/S0169-7439(03)00094-7.," *comparative study of K-nearest neighbour,* 2003. |
| [20] | Y. L. Y. Z. M. G. S. Wan, "Deep multi-layer perceptron classifier," *For behavior analysis to estimate Parkinson’s disease severity using smartphones, IEEE Access 6 (2018) 36825–36833, http://dx.doi.org/10.1109/ACCESS.,* 2018. |
| [21] | L.E. Peterson, "K-nearest neighbor, Scholarpedia," 2009. |
| [22] | H. H. P.H. Swain, "Decision tree classifier: Design and potential," *IEEE Trans. Geosci. Electron. GE-15 (3) (1977) 142–147, http://dx.doi.org/10.1109/tge.1977.6498972.* |
| [23] | K. Murphy, " Naive Bayes classifiers generative classifiers," *Bernoulli 4701 http://dx.doi.org/10.1007/978-3-540-74958-5\_35,* pp. 1-8, 2006. |
| [24] | E. K. G. K. K. N. T. Rymarczyk, "Logistic Regression," *machine learning in process tomography, Sensors (Switzerland) 19 (15) http://dx.doi.org/10.3390/s19153400.,* 2019. |
| [25] | A. K. A. Natekin, "Gradient boosting machines,," *a tutorial, Front. Neurorobot. http://dx.doi.org/10.3389/fnbot.2013.00021.,* 2013. |
| [26] | R. Rojas, "a tutorial introduction to adaptive boosting." Freie University, Berlin, Tech. Rep 1.1," *AdaBoost and the super bowl of classifiers,* pp. 1-6, 2009. |
| [27] | P. Nieminen, " multilayer perceptron neural networks, Training," 2010. |
| [28] | B. G. D. Ruta, "Classifier selection for majority voting," *Inf. Fusion 6 (1),* pp. 63-81, 2006. |
| [29] | I. Levner, "Feature selection and nearest centroid classification for protein mass," *spectrometry, BMC Bioinformatics ,* pp. 1-14, 2005. |
| [30] | S. I. Gallant, "Perceptron-based learning algorithms.," *IEEE Transactions on neural networks 1.2 ,* pp. 179-191, 1990. |
| [31] | N. Biswas, " comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach," *Healthcare Analytics ,* 2022. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |