National College of Ireland

Stroke Detection and Prediction Using Deep Learning Techniques and Machine Learning Algorithms

MSc Research Project
MSc in Data Analytics

Ripu Murdhan Chandramohan
Student ID: x20186673

School of Computing
National College of Ireland

Supervisor: Qurrat Ul Ain

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ripu Murdhan Chandramohan

Student ID: x20186673

Programme: MSc in Data Analytics **Year:** 2022

Module: Research Project

Supervisor: Qurrat Ul Ain

Submission Due Date: 16-09-2022

Project Title: Stroke Detection and Prediction Using Deep Learning Techniques and Machine Learning Algorithms

Word Count: 8048 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Ripu Murdhan Chandramohan

Date: 16/09/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Stroke Detection and Prediction Using Deep Learning Techniques and Machine Learning Algorithms

Ripu Murdhan Chandramohan

x20186673

Abstract

A stroke is one of the major causes of mortality in the World, contributing for the death of more individuals in each and every year. The medical industry has made significant strides in curing strokes; nonetheless, a stroke can occur at any time, and its rate of damage is so high that even if it is treated, it can still result in lifelong disability. Using datasets that are available to the public, the purpose of this study is to identify patients who are at risk of having a stroke. This may be done by constructing six separate categorization models as predictors. The Six classification techniques are assessed using two distinct sampling approaches i.e., Adaptive Synthetic Sampling (ADASYN) and Synthetic Minority Oversampling Technique (SMOTE) since medical datasets tend to be very unbalanced. Based on the assessments and conclusions provided in the report, SMOTE and ADASYN fared identically on Accuracy (except for the Neural Network model, where ADASYN did somewhat better than SMOTE). Similarly, approaches performed fairly similarly for Random Forest (SMOTE Accuracy = 76.9, ADASYN Accuracy =77.3), Decision Tree (SMOTE Accuracy = 54.6, ADASYN Accuracy = 56.5), Adaboost (SMOTE Accuracy =55.8, ADASYN Accuracy=55.2), SVC (SMOTE Accuracy =50, ADASYN Accuracy =50). The acquired results are encouraging and have effectively helped towards the solution of the stroke detection problem in the medical business.

1 Introduction

The health industry is one of the most crucial since it concerns with people's lives. The medical field has identified remedies for some life-threatening diseases; but, even after treatment, certain illnesses may cause long-term incapacity. Furthermore, such illnesses demand immediate medical attention due to their high harmful frequency to the human body, and such diseases can go unrecognized due to the lack or very infrequent incidence of symptoms. Stroke is one of these disorders.

1.1 Background and Motivation

Stroke occurs when blood flow to specific areas of the brain is disrupted or diminished, resulting in the death of cells in those areas of the brain caused by a lack of oxygen and nutrients. According to the World Health Organization, stroke is the world's second major cause of death, responsible for approximately 11% of all mortality. The World Health Organization (WHO) estimates that 15 million people are suffering from strokes annually, with one person dying every four to five minutes. Stroke is now classified as a worldwide phenomenon. Every fourth person over the age of 25 will encounter a stroke at a certain time

in life. One of the most prevalent disease processes that leads to stroke is atherosclerosis. Strokes become much more common as individuals age[1].

So far, the work done to detect patients at risk of stroke has been focused on the use of equipment or forecasting based on medical examination reports such as Magnetic Resonance Imaging (MRI). Ones who believe they are healthy will not undergo such checks, decreasing the effectiveness of such measures [2]. As a result, the focus of this scientific report is on identifying patients at risk of stroke using the capabilities of machine learning and deep learning algorithm. Having said that, Several Deep Learning and Machine Learning methods were effectively employed in (Tazin et al.,2021). In this study, the similar approach will be used with multiple deep learning and machine learning methods, as well as multiple sampling approaches.

1.2 Research Question

As in the actual world, the percentage of patients at risk of stroke is projected to be significantly less than the total patients registered in the majority or practically all hospitals. As a result, the dataset is always expected to be severely skewed, with stroke patients being the primary emphasis. As a response, the purpose of this research is on stroke detection and prediction using deep learning and machine learning algorithms with a variety of sampling strategies. Therefore, this study proposal will address the following research questions.

1. *"How well do SVM, Bagging and Adaboost predict stroke using Adaptive Synthetic Sampling (ADASYN) in addition to Neural network?"*

2. *How well does Adaptive Synthetic Sampling (ADASYN) outperform SMOTE Sampling Techniques for Stroke Prediction?*

1.3 Research Objectives

The primary purpose of this research is to develop a method to assist the typical person in assessing the likelihood of having a stroke, as well as acquiring particularly worrisome intervention and recovering control over the situation following the event. The sooner a stroke victim is treated, the higher his or her chances of survival. As an outcome, this research will be helpful in the medical field and therefore would help to answer the question, "How is computing technology beneficial in the medical world?"

A review of literature, research techniques, design requirements, and evaluating metrics are included in the following sections of the reports. As a result, there is a conclusion.

2 Related Work

This section lists and analyses several written works and study. A full review of stroke identification techniques, as well as major machine learning and deep learning methodologies required to have a deeper understanding of this research endeavour. Many literary reviews in this field are divided into distinct categories, such as,

1. Sampling and Feature Techniques Used in Previous Work to Handle Imbalanced Datasets
2. Using Sampling Techniques, Previous Research looked at which Machine Learning and Deep Learning methods work better
3. Various Imbalanced Datasets Used for Stroke Prediction
4. Conclusion

2.1 Sampling and Feature Techniques Used in Previous Work to Handle Imbalanced Datasets

Sampling strategies are by far the most popular method for dealing with data that is not evenly distributed. Over-sampling and under-sampling are two forms of samplings. Various methods of over- and under-sampling are available. This section, therefore, tries to thoroughly evaluate the various sampling tactics adopted in past research.

The hybrid sampling methodology, which takes into account both over and under sampling, is another sampling strategy. Fuzzy Distance-based Under sampling (FDUS) and SMOTE techniques were utilized by researchers in (Zorkeflee, M., et al., 2015) to enhance classification performance when dealing with very unbalanced data. The FDUS method of under sampling eliminates selected samples from the majority class by using fuzzy logic. While FDUS reduces bias in choosing samples that should be removed, SMOTE eliminates overfitting. These advantages make it possible for the suggested FDUS+SMOTE to enhance classification performance. Performance of the proposed method is compared to SMOTE+Tomek and SMOTE+ENN, two combination techniques. Similar to this, researchers in (Zeng, M et al., 2016) used SMOTE with Tomek links to balance the data during the pre-processing stages. The authors used a number of machine learning strategies after pre-processing. The analysis showed that evaluation metrics have consistently improved when compared to SMOTE alone in terms of illness prediction. studies by (Alharbi, F et al., 2022). Concatenating synthetic training data generated by the various sample approaches was accomplished by using a combination of distance-based method (DBM), noise detection-based method (NDBM), and cluster-based method (CBM), as described in Comparing Sampling Strategies for Unbalanced Data. DBM is the result of combining SMOTE with Random SMOTE. Modified synthetic minority oversampling method (MSMOTE) and SMOTE Tomek connections are both included into NDBM, which uses a hybrid approach. Cluster-Based Synthetic Oversampling (CBSO) and Proximity Weighted Synthetic Oversampling Technique (ProWSyn) are combined in CBM In terms of training time, the DBM approach outperformed the NDBM and CBM methods.

(Rajora M et al.,2021) research efforts prompted me to do stroke prediction study. Once the data for exploratory data analysis has been imported, the spark program is started. The SMOTE approach was used instead of blatantly oversampling the data to account for any potential class imbalance. A random point is chosen from the minority class, and then a search is conducted for the point's nearest neighbour, which is the next most distant neighbour. An artificial example is now constructed between the two places by choosing one of their neighbours. Some columns were eliminated and unwanted noise was reduced throughout the study's pre-processing phases (Tazin, T et al.,2021). Label encoding was used to convert the strings into numbers. A large portion of the dataset utilized to make stroke predictions is significantly out of balance. As a result, forecasts and analyses will be wrong

when uneven data is used incorrectly. Unbalanced data must thus be addressed first in order to build an effective model. In order to do this, they used the SMOTE method. Pre-processing of very imbalanced datasets required the researchers in (Sailasya G et al., 2021) to remove a few columns and seek for null values. Label encoding was used to encode strings into numbers. In order to deal with the imbalanced data, they used the under-sampling approach. Under-sampling the majority class in order to match the under-sampling of the minority class ensures that the data is balanced. The class with the value "0" is under sampled when compared to the class whose value is "1. In their research, (Menezes, L. et al.,2021) they used SMOTE (Synthetic Minority Oversampling Approach) sampling technique using the imblearn package of python to accomplish the same. The MinMaxScaler function was then used to scale the entire dataset from 0 to 1. After pre-processing the datasets in the EDA step, researchers in (M. M. Islam et al.,2021) and (Al-Islam et al.,2021) used the synthetic minority over sampling approach (SMOTE) to deal with imbalanced data. This computation aids in resolving the issue of over-fitting brought on by random oversampling.

From what has been said so far, it is feasible to draw the conclusion that while Hybrid sampling has a potential to offer more balanced results than both Over- and Under-sampling, it cannot guarantee superior outcomes because the performance of sampling method is only data dependent.

2.2 Using Sampling Techniques, Previous Research looked at which Machine Learning and Deep Learning methods work better

Class names in medical databases tend to be uneven since the number of patients at risk is expected to be significantly smaller than the total number of people recorded. Machine learning and deep learning approaches are heavily utilized in the prediction of illnesses. In further depth, this portion of the study addresses which sampling methodology works best for the machine learning and deep learning algorithms to predict strokes in a variety of experiments.

Random Forest, Decision Tree and Voting Classifier and Logistic Regression were some of the machine learning models utilized by the researchers in their prediction utilizing SMOTE sampling (Tazin T et al., 2021). Each algorithm's result has acceptable accuracy. The logistic regression model did poorly when compared to the other alternatives.

To better predict strokes, researchers (Rakshit, T et al., 2021) used a variety of machine learning models. Random Forest, Logistic Regression, KNN, Decision Tree, and Naive Bayes were some of the models utilized. Accuracy score, Precision (P), Recall (R), and F-measure were all employed to evaluate the algorithm's performance in this study. The confusion matrix was used to make these measurements. According to the data gathered after applying all of the models, the Decision Tree algorithm had the best prediction accuracy of 100 %, while Nave Bayes had the poorest prediction accuracy of 86.84 %.

SMOTE sampling was employed in the research study by the authors to apply models such as logistic regression, Gaussian Naive Bayes, and ANN (Artificial Neural Network) (Menezes L et al., 2021). An average step size of one second per iteration and a loss score of 0.2355 during the last 400 iterations of the training dataset yielded a 90% accuracy for the ANN

model, which was then implemented. Prevents overfitting by limiting training to 400 of 800 epochs The test dataset had an accuracy of 89% and a loss score of 0.2719, whereas logistic regression was 82% accurate.
A random down sampling strategy was utilized to mitigate the negative effects of an imbalanced dataset in a study on predictive analytics for stroke prediction (Dev, S., Wang et al., 2022). To make stroke predictions, they employed a combination of neural networks, decision trees, and random forests (RF). In order to develop a convolutional neural network (CNN), the first two layers were convolutional and the last two were linear. Neuronal networks surpassed Random Forests and Decision Trees with a prediction accuracy of 77%, according to the study's findings.
Other researchers (M. Jalaja Jayalakshmi and colleagues, 2021) used machine learning to analyse and forecast data. J48, Bayes Network Classifier, Naive Bayes Classifier, and Adaboost were some of the models that were used in the experiment. When compared to the Bayes Network and Nave Bayes classifiers, the AdaBoost and J48 algorithms correctly classify the data 95.69 percent of the time.

Another study (Al-Islam, F et al., 2021) used the Logistic Regression Classifier, Random Forest Classifier, and XGBoost Classifier models to try to predict strokes in participants. SMOTE sampling was utilized to address the imbalanced data before the models were applied. Following SMOTE, the results show that random forest outperformed the other two models, with an accuracy rate of 99.9%.
According to another study (GholamAzad, M et al., 2021), the likelihood of stroke was predicted using a logistic regression model. In their investigation, logistic regression properly predicted 5,091 people who were at risk for a stroke with 100% accuracy. Overall, the prediction error was 5.89 percent.
In a separate study, researchers employed logistic regression, SVM, and C5 to predict mortality in stroke patients using over-sampling, under-sampling, and SMOTE sampling strategies (Hadianfard, Z et al., 2022). They selected these models in order to gain experience. Accuracy, sensitivity, and specificity, as well as AUC and ROC curves and kappa statistics, were all assessed for the prediction models employed in the study.

2.3 Various Imbalanced Datasets Used for Stroke Prediction

Authentic datasets are imbalanced by nature. The term "imbalanced data" is used to describe categorization problems in which certain categories have a higher or lower percentage of the total data. We'll talk about the numerous imbalanced datasets utilized in stroke prediction research in this part as a result.

Tazin, T. and colleagues (2021) used publicly available datasets in their research. There are 5110 rows in total throughout the dataset's 12 columns. Stroke risk has not been recognized if the value is 0, but it has been discovered if the number is 1. Only 249 entries in the stroke column have a value of 1, whereas 4861 items in the column have a value of 0. It's common practice to use pre-processing to balance and increase the accuracy of data. A SMOTE approach was utilized before any models were used because of this.
To conduct their research, (Menezes, L. et al., 2021) used a HealthCare dataset that had relevant patient data that was easily accessible. Stroke victims were 4861, while 249 people

are at high risk of having one. To get satisfactory results, the researchers added an alcohol consumption column to an existing dataset and converted hypertension measurements to continuous values. Age, heart disease and diabetes, as well as pre-existing bp readings of 0 or 1, were all taken into consideration by the researchers. SMOTE was used to ensure that the dataset was evenly split between Stroke and No Stroke before the models were applied.

The researchers employed electronic medical records in their study (Dev, S., Wang, H., et al., 2022). (EMR). The health status of a patient may be tracked down using this computer-readable record. Vital signs, diagnosis, and examination findings can all be included in a patient's medical records. Researchers used McKinsey & Company's electronic health records to perform their investigation. A total of 29072 patients' electronic health records (EHRs) are included in the collection. Output features and input attributes are totalled to come up with just one. It was owing to ethical concerns that they did not include the patient's name in their study, which included 10 input characteristics and one response variable. Random sampling is used to maintain an even distribution of stroke labels across the dataset.
The research employed two datasets with identical baseline characteristics (M. M. Islam et al., 2021). NCC data sets derived from the H-type Hypertension and Stroke Prevention and Control Project (HSPCP) as well as the China Stroke Primary Prevention Trial (CSPPT) were both analysed in the study.
Participants aged 0 to 17 or those who had not yet been diagnosed with stroke were excluded from the research. The data collected from patients (GholamAzad, M et al., 2022) were both quantitative and qualitative in character.
The Imam Reza Hospital of Urmia in Iran and the neurology departments of the Imam Khomeini Teaching Hospital in Iran were employed in their study (Hadianfard, Z. et al., 2022). There was an examination of medical records using the ICD-10 categorization system (International Classification of Diseases and Related Health Problems). Over-sampling, under-sampling, and SMOTE were used to balance datasets main variables.

2.4 Conclusion

As a result of the literature review utilized for stroke prediction, the following limitations have been recognized to my knowledge and addressed in this study.
SMOTE, under sampling, and oversampling were among the sampling procedures utilized by the majority of the researchers, according to section 2.1. SMOTE-Tomek links which are the most often used hybrid sampling approach, has been used by a very limited number of other persons, as well.
Deep Learning methods like CNN and ANN were employed in the majority of research, as described in section 2.2. In this study, I'll investigate neural networks method utilizing ADASYN and SMOTE sampling approaches. As recommended by current studies in the publication (Tazin, T et al.,2021), I'll also investigate other machine learning techniques, such as Adaboost, Support Vector Machine and bagging.
As a result of the unbalanced data, this study will employ several sampling strategies, such as ADASYN (adaptive synthetic sampling) and SMOTE (oversampling). These sampling strategies will be utilized in the application of machine learning and deep learning techniques such as Random Forest, Decision Tree classifier, AdaBoost, SVM, Bagging, and neural networks. These models are evaluated based on their performance using a variety of measures, including confusion matrix, sensitivity, F1-score, precision and accuracy.

3 Applied Experimental Methodology

Data mining is practically limitless until a conclusion is derived from raw data. Universally acknowledged methodologies and procedures are needed to track all processes. Sample Explore Modify Model and Access (SEMMA), Knowledge Discovery and Data Mining (KDD) and Cross Industry Standard Process for Data Mining (CRISP-DM) are typical analytics methodologies. This project has no business layer deployment; hence KDD was used to diagnose stroke-prone individuals (Shafique, U et al.,2014).

3.1 KDD (Knowledge Discovery and Data Mining) technique

The Knowledge Discovery and Data Mining (KDD) technique is data-focused; hence it's applied in this project. KDD is a data-mining approach for uncovering new information. There are five distinct processes in KDD, which are outlined below in the perspective of stroke prediction.

1. **The Data Selection Phase**
   Methods for selecting and obtaining relevant data from the data collection are discussed in this step. The dataset was gathered using Kaggle[1] in this instance. Its result is the data that is fed into the data pre-processing stage, which in turn feeds the next phase.

2. **The Data Preparation Phase**
   The data pre-processing phase involves operations like as cleaning up missing values using data processing tools, cleaning up noisy data, and selecting feature sets that will be used in the analysis, I applied Label Encoder () and Standard Scalar () functions in this study. The output of this phase is sent into the stage of data transformation.

3. **The Data Transformation Phase**
   To facilitate data mining, data transformation involves converting data from one format to another. As part of this research, sklearn. pre-processing is used to do data standardization and label encoding on the data. Python's OneHotEncoder and StandardScaler libraries were utilized. As a result, the output of this phase is sent into the stage of data mining.

4. **The Data Mining Phase**
   This is the most essential stage, when data mining methods are employed to detect patterns in a dataset. Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling (ADASYN) are two different data sampling strategies that were utilized to create the six different categorization models in this study. Following this stage, the patterns found in the data are assessed as a part of the assessment process.

5. **The Evaluation / Interpretation Phase**
   The trained models are assessed based on confusion matrix, sensitivity, F1-score, precision, and accuracy. The final phase of KDD yields knowledge that may be used to anticipate strokes in the future. Figure 1 depicts the whole cycle of KDD approach, as seen here.

[1] https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

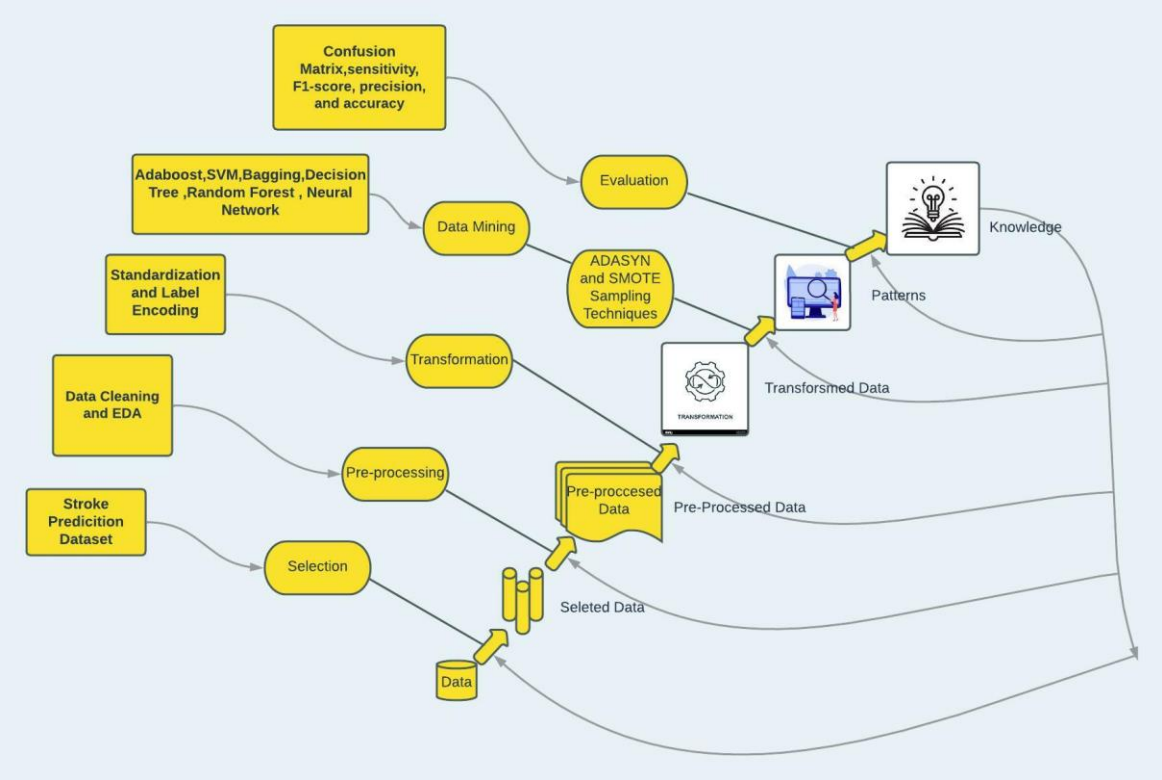

Figure 1: Flowchart for a KDD

4 Design Specification

Specifications for a project's design summarize the project's general architecture, as well as details such as the work process, methods, technologies, and approaches that will be utilized to accomplish the project. Any analytics project's architectural design may be characterized as either a two- or three-layered architecture design. Three-tier architecture is used in this study, including a Data Transition layer, Application layer, and Output layer. Data may move more easily across the various layers since they are all linked together.

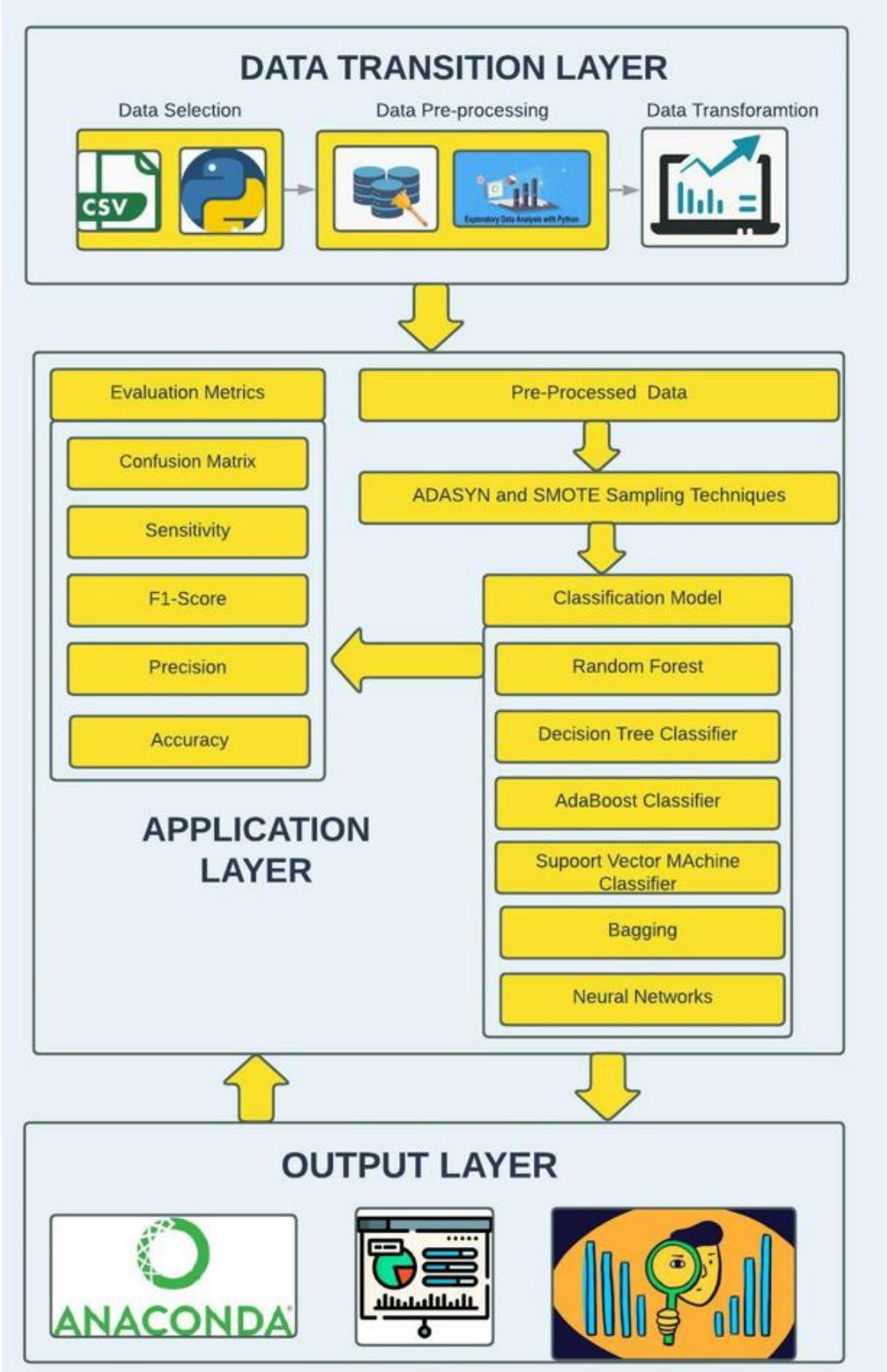

Figure 2: 3-Layer Project Plan

4.1 The Data Transition Layer

Using Python, this layer performs actions such as collecting data, data verification, data transformation with label encoders, and standardization before transmitting the transformed data to the next Layer.

4.2 The Application Layer

This is where all the application logic is done. The ADASYN and SMOTE sampling techniques are used to apply six unique classification models once the input has been pre-processed. These models are then analysed using the assessment metrics.

4.3 The Output Layer

This is the sole layer that the user can see, and it is responsible for receiving input from the client and transmitting output to the client. The results are presented in data visualizations created in Anaconda as well as in Microsoft Excel.

In conclusion, taking into consideration the objective of this project, the combination of KDD and 3-Tier Design was selected to implement a strategy that is more data-centric.

5 Stroke Detection Model Implementation, Evaluation, and Results

The implementation, assessment, and outcomes of the six separate models, each built using a different sampling approach, are all covered in this chapter. It also talks about the things that were done to prepare for the deployment, like exploratory data analysis, data cleaning, pre-processing, feature selection, and data scaling.

5.1 Implementation Resources

The scripts for this project were developed in Jupyter notebook, and the execution of this project and the visualization of the data were both accomplished with the help of the Anaconda Python package.

5.2 Sampling Approaches

After steps like as data preparation and the extraction of features have been finished, the dataset is next segmented into training data and test data with an 80:20 split between the two types of data. On the basis of the evaluation conducted in Section 2.1, a number of different pattern methods were selected to apply to the training data. These are the

1. **Synthetic Minority Oversampling Technique (SMOTE)**
   SMOTE provides synthetic minority class instances for class balancing. SMOTE picks close samples in the feature space, draws a line between them, and creates a new sample along that line. The SMOTE () method of the imblearn. over sampling package in Python was used to implement it on the training dataset.

2. **Adaptive Synthetic Sampling (ADASYN)**
   ADASYN is a technique for oversampling. It's a higher model of SMOTE. This is basically another SMOTE extension that creates false samples inversely proportional to minority class density. It is aimed to construct fictitious times in areas of the typical area with a low density of minority cases and few or none in areas with a high density. The ADASYN () method in the imblearn. over sampling package in Python was used to implement it on the training dataset.

5.3 Measures of Stroke Detection Prediction Model

The assessment was carried out by splitting the real dataset in a divided 70:30 ratio for training and testing, utilizing the 30% test data produced during data sampling operations. Metrics such as the confusion matrix, sensitivity, F1-score, precision and accuracy were considered throughout the evaluation process for this project. As a result of the class imbalance, accuracy is no longer recommended for usage as a performance indicator. So, it is utilized as a complement to the other indicators and not as a primary factor in making decisions.

1. **Confusion Matrix**
   There are four types of predictions that are summarized in the Confusion Matrix: true positive, true negative, and false positive. True Positives (stroke cases were accurately predicted), True Negatives (Healthy instances were successfully predicted), and false positive and false negative (Cases of stroke misdiagnosed as healthy). The confusion matrix is the single most important performance element. where TN stands for true negatives, FN for false negatives, FP for false positives, and TP for true positives.

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

2. **Sensitivity (Recall)**
   This is the exact number of stroke cases compared to the overall number of stroke cases. The computation is performed using the formula shown below:

$$\text{Sensitivity} = \frac{(TP)}{(TP + FN)}$$

3. **Accuracy**
   A specific breakdown of stroke and healthy patients in the overall forecasts is predicted to look like this. The computation is performed using the formula shown below:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

4. **F1-score**
   It's possible to combine Precision and Recall into one unit known as the F1-score. It's basically a symphonic medium between precision and recall. In the F1-score, the influence of both measures is equal.

5. **Precision**
   Classification accuracy is a percentage of the total number of cases categorized in that particular classification. Classifiers may only categorize an instance as belonging to a given class if the instance is indeed a member of that class itself.

$$Precision = TP / (TP + FP)$$

5.4 Data Exploration, Data Pre-processing and Feature Extraction

In this section, we'll go over all of the steps that were taken before we can use Machine Learning Algorithm.

5.4.1 Dataset Description

In this experiment, we used a stroke prediction dataset that is publicly available on Kaggle. There are more than 5000 rows and 12 columns in this dataset. Only one of two values can be used to represent the output column stroke: 1 or 0. As an example, 0 indicates that there is no danger of stroke, but 1 indicates that there is. This study will make use of multiple sampling procedures, such as Adaptive Synthetic Sampling (ADASYN) and Synthetic Minority Oversampling Technique (SMOTE), to ensure that the dataset is balanced due to the vulnerability of medical records to class distribution issues.

5.4.2 Data Exploration

A thorough understanding of the stroke prediction dataset necessitates exploratory data analysis. During this phase, all variables, their class balance or distribution, as well as their connection to each other, will be analysed. A lot of analyses and visualisations are involved at this step, but this article focuses on a few of the most significant ones and others that aren't will be included in the configuration document.

While attempting to analyse the class distribution of the dependent variable (Stroke), as shown in the visual analysis in Figure 3, it was discovered that the dataset is highly imbalanced, with only 249 cases, which corresponds to 4.87 percent of the dataset for stroke, being reported against the 4861cases, which corresponds to 95.1 percent instances of healthy cases.

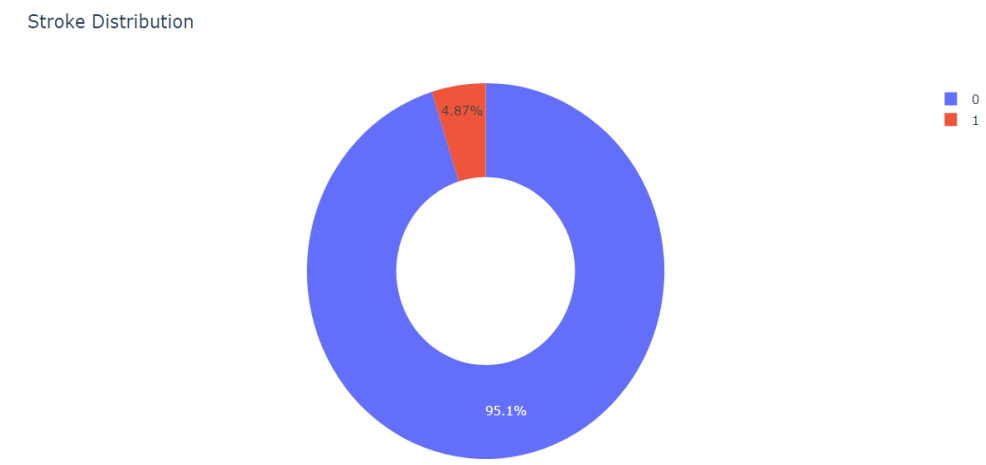

Figure 3: Distribution of Stroke

An equivalent analysis was carried out for additional categorical variables, such as the distribution of strokes according to gender, strokes according to hypertension, strokes according to heart disease, strokes according to work type, and smoking status distribution.

For further data analysis, a distribution map was generated in the case of continuous variables such as age, the body mass index (BMI), and the average glucose level by utilizing the distplot () function of the seaborn package in the Python programming language.

The distribution of ages across the sample is relatively uniform, as seen by Figure 4, which highlights this point. Figures also imply that a person's risk of having a stroke increase with age, with the likelihood of having a stroke increasing with age beyond the age of 40 as compared to those under the age of 40.

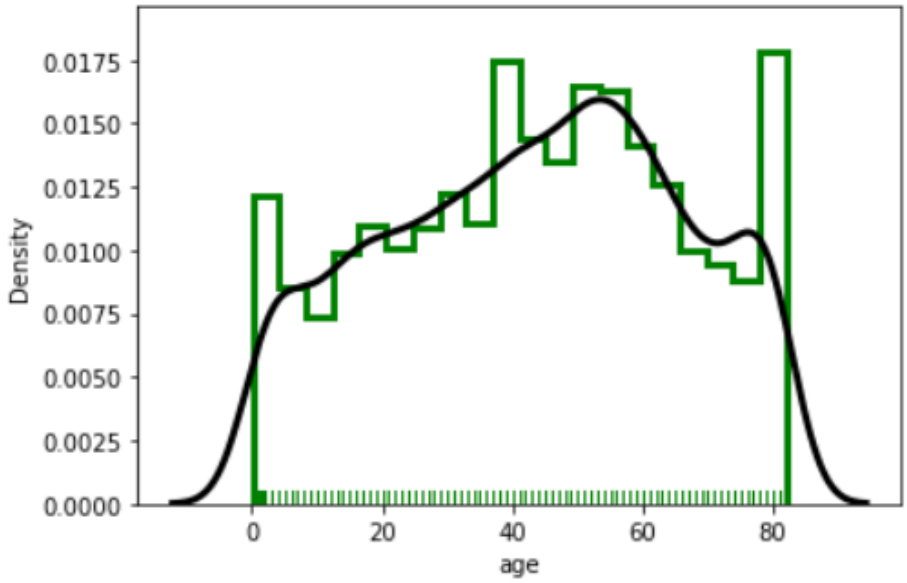

Figure 4: Distribution of Age

In addition to determining relations and assessing the quality of the data, one of the primary goals of exploratory data analysis was to locate inconsistencies and null values in the dataset. This was an extremely significant aspect of the process. During the process of utilizing Python to investigate the missing data,

it was discovered that, as shown in Figure 5, there were 201 missing values associated with BMI.

```
stroke_data.isnull().sum()
```

```
gender               0
age                  0
hypertension         0
heart_disease        0
ever_married         0
work_type            0
Residence_type       0
avg_glucose_level    0
bmi                201
smoking_status       0
stroke               0
dtype: int64
```

Figure 5: Dataset Missing Values

5.4.3 Data Pre-Processing and Feature Selection

Raw data may be transformed into useful information for Machine Learning models by utilizing data processing and feature selection. The following are some of the most significant things that were accomplished during this phase:

1. Null Values Imputation

Following the completion of the Exploratory Data Analysis, it was determined that the bmi column included a total of 201 null values; hence, Python was used to impute these values using the fillna function while passing in the median value.

2. Categorical Variable Encoding

It is necessary to encode categorical variables as integers for the sake of making machine learning models understand them. Using the LabelEncoder () method that is included in the sklearn. preprocessing package in Python, encoding was carried out on categorical variables.

3. Multicollinearity and Feature Scaling

It is necessary to do Feature Selection to ensure that only appropriate features are provided to the machine learning models in order to increase performance and avoid difficulties with modelling. In addition to this, correlations between the characteristics were found in order to rule out the possibility of multicollinearity. It is possible to draw the conclusion, after looking at the visualization in Figure 6, that multicollinearity does not exist in the characteristics that were chosen. Python's StandardScalar () function, which is part of the sklearn package, was utilized in order to scale the data for the continuous variables.

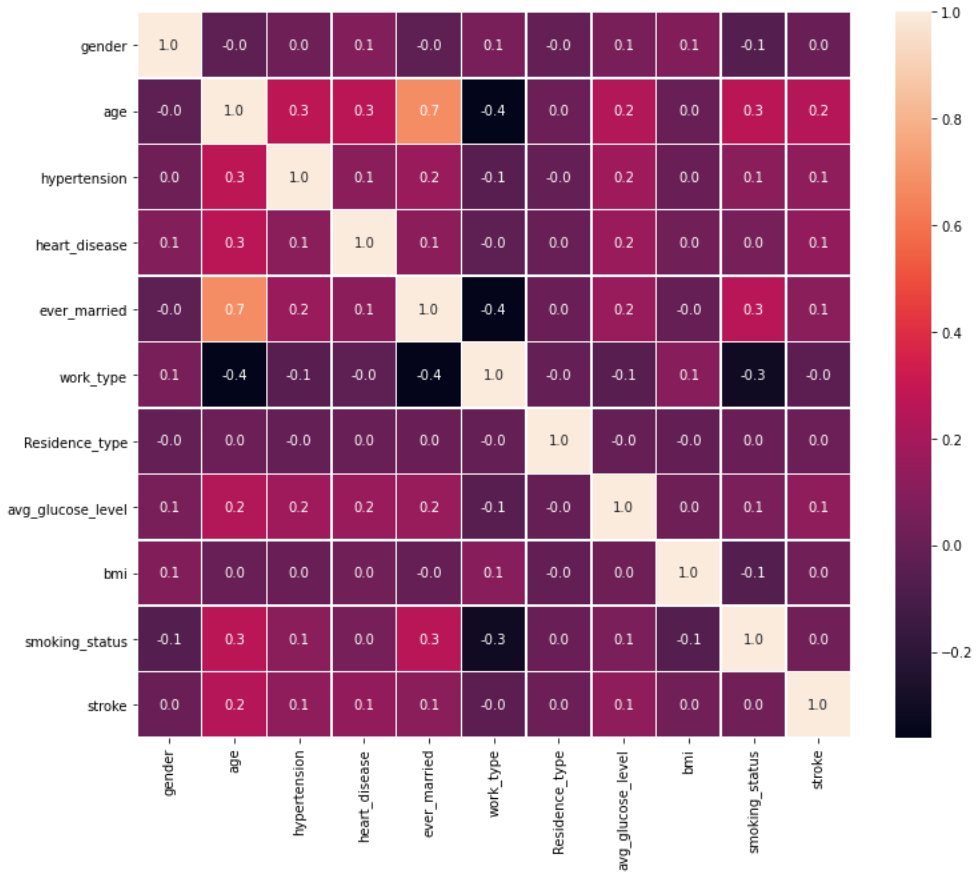

Figure 6: Stroke Correlation Plot

5.5 Random Forest Implementation, Evaluation and Results

Random forest predicts classification and regression problems. Random Forest is an ensemble of decision trees using random attributes and data samples. Random Forest is the mode of all tree outcomes. Random Forest has a lower error range and is less prone to overfitting, hence it's vital to this research.

sklearn.ensemble package's RandomForestClassifier() method was used to construct Random Forest for both of the sample strategies.

Results:

The results of Random Forest are displayed in Table 1 with two different sample techniques and also without sampling strategy. Comparing the model's performance to Accuracy measures demonstrates its ability to distinguish between stroke and healthy instances. These metrics demonstrate that sampling techniques and also without sampling strategy have achieved almost comparable accuracy. As opposed to ADASYN's 1456 =True Positives, SMOTE's 1453 =True Positives, and Without sampling technique's 1452 =True Positives. Thus, while using ADASYN, Random Forest yielded more accurate true positives than SMOTE sampling and random forest without sampling approach.

Serial No	Model Name	Sampling Technique	Accuracy	Precision	Recall	F1-Sore	TP	FN	FP	TN
1	Random Forest	Without Sampling Technique	0.774	0.95	1	0.77	1452	75	5	1

2	Random Forest	SMOTE	0.773	0.95	1	0.97	1453	74	4	2
3	Random Forest	ADASYN	0.769	0.95	1	0.97	1456	74	1	2

Table 1: Random Forest Performance using Different Sampling Techniques

5.6 Decision Tree Implementation, Evaluation and Results

The Decision Tree classifier model is a sequential process in which a condition is assessed as a node, and its consequences are separated into many branches. The process repeats based on the parameters that have been specified until it terminates with leaves that forecast the result of the stroke detection.

Using the DecisionTreeClassifier() method that is included in the sklearn.tree package in Python, a Decision Tree was built for each of the two different sample strategies.

Results:

Table 2 shows the Decision Tree Classifier's performance using two sampling methods and no sampling approach at all. The model's capacity to differentiate between stroke and healthy cases is demonstrated by comparing its performance to Accuracy measurements. These measures show that the accuracy attained with and without a sampling procedure is quite close. 1399 =True Positives with accuracy of 56.5 for ADASYN; 1401 =True Positives with accuracy of 54.6 for SMOTE; and 1395 =True Positives with accuracy of 56.4 for Without sampling procedure. Since SMOTE sampling and the Decision Tree without sampling strategy both underperformed compared to Decision Tree when utilizing ADASYN, it follows that Decision Tree produced more accurate true positives with improved accuracy.

Serial No	Model Name	Sampling Technique	Accuracy	Precision	Recall	F1-Sore	TP	FN	FP	TN
1	Decision Tree	Without Sampling Technique	0.564	0.96	0.96	0.96	1395	62	63	13
2	Decision Tree	SMOTE	0.546	0.96	0.96	0.96	1401	66	56	10
3	Decision Tree	ADASYN	0.565	0.96	0.96	0.96	1399	58	63	13

Table 2: Decision Tree Performance using Different Sampling Techniques

5.7 AdaBoost Implementation, Evaluation and Results

It's an ensemble-boosting model called AdaBoost, and it combines several weak classifiers to get a powerful one that may be used for classification. AdaBoost then prioritizes the best-performing models to enhance overall results.

The sklearn.ensemble package's AdaBoostClassifier() function was used to implement the AdaBoost classifier in Python for each of the two sampling methods.

Results:

The results of the Adaboost Classifier utilizing two different sampling strategies and no sampling strategy are displayed in Table 3. Accuracy comparisons show that the model can correctly identify stroke patients from healthy controls. This data demonstrates that the

precision achieved with and without a sampling technique is very similar. 1398 = ADASYN True Positives; 1396 = SMOTE True Positives with; 1397 = ADASYN True Positives Without Sampling. It follows that Adaboost generated more accurate true positives compared to SMOTE sampling and the Adaboost without sampling method while using ADASYN.

Serial No	Model Name	Sampling Technique	Accuracy	Precision	Recall	F1-Sore	TP	FN	FP	TN
1	Adaboost	Without Sampling Technique	0.558	0.96	0.96	0.96	1397	64	60	12
2	Adaboost	SMOTE	0.558	0.96	0.96	0.96	1396	64	61	12
3	Adaboost	ADASYN	0.552	0.96	0.96	0.96	1398	65	59	11

Table 3: AdaBoost Performance using Different Sampling Techniques

5.8 Support Vector Classifier Implementation, Evaluation and Results

A Support Vector Classifier, often known as SVC, is a model that differentiates between the results of a classification by employing the hyperplane.

By utilizing the sklearn.svm package in Python and its SVC() method, an SVC model with a Linear kernel was constructed for each of the two sampling strategies.

Results:

Table 4 presents the findings obtained by the SVC with the use of two distinct sampling procedures in addition to the absence of any sampling approach. The accuracy comparisons demonstrate that the model is able to accurately differentiate between healthy controls and stroke patients. The results presented here show that the precision that can be reached with or without the use of a sampling approach is extremely close to 0.5. As a result, it may be deduced that all of the many ways in which SVC performed were identical to one another.

Serial No	Model Name	Sampling Technique	Accuracy	Precision	Recall	F1-Sore	TP	FN	FP	TN
1	SVC	Without Sampling Technique	0.5	0.95	1	0.97	1457	76	0	0
2	SVC	SMOTE	0.5	0.95	1	0.97	1457	76	0	0
3	SVC	ADASYN	0.5	0.95	1	0.97	1457	76	0	0

Table 4: Support Vector Classifier Performance using Different Sampling Techniques

5.9 Bagging Classifier Implementation, Evaluation and Results

As a way to reduce the variance in an uncertain dataset, bagging is an ensemble learning technique. The term "bootstrap aggregation" refers to the same thing. Data points can be selected several times when using bagging, which takes a random sample of training data and

replaces it. In order to improve the performance of these models, they must be trained separately for each type of task (regression or classification, for example).
Using the BaggingClassifier () method included in the sklearn. ensemble package in Python, bagging was applied for each of the two different sample approaches.

Results:

In Table 5, we see the results of the Bagging test conducted with three different sampling strategies (no sampling, simple random sampling, and systematic sampling). Comparative analyses of the model's performance show that it can reliably identify stroke patients from healthy controls. According to these findings, a precision of 0.73 is quite near to being achievable both with and without resorting to a sampling strategy. This suggests that the various modes of Bagging's performance were all equivalent.

Serial No	Model Name	Sampling Technique	Accuracy	Precision	Recall	F1-Sore	TP	FN	FP	TN
1	Bagging	Without Sampling Technique	0.73	0.95	1	0.97	1457	75	0	1
2	Bagging	SMOTE	0.73	0.95	1	0.97	1457	75	0	1
3	Bagging	ADASYN	0.73	0.95	1	0.97	1457	75	0	1

Table 5: Bagging Classifier Performance using Different Sampling Techniques

5.10 Neural Network Implementation, Evaluation and Results

The Neural Network model is composed of neurons distributed over numerous layers. These neurons comprehend the behaviour of the data in order to recognize the underlying pattern, which is subsequently utilized for predictions.
Using the KerasClassifier() method included in the keras.wrappers.scikit learn package in Python, a Neural Network was built for each of the two different sampling strategies. After fine-tuning the settings, the best results were obtained while using the sklearn package with epochs equal to 100 and batch size equal to 20.

Results:

Table 6 shows the output of the Neural Network with and without a sampling method. The model's ability to tell the difference between stroke and healthy cases is shown when compared to Accuracy measurements. In contrast to the 3404 =True Positives obtained using ADASYN, the 3400 =True Positives obtained using the SMOTE method, and the 3403 =True Positives obtained using no sampling method. The results of the ADASYN study showed that the Neural Network outperformed the SMOTE sampling method and the Neural Network without sampling method in terms of the number of correct positives it produced.

Serial No	Model Name	Sampling Technique	Accuracy	Precision	Recall	F1-Sore	TP	FN	FP	TN
1	Neural Network	Without Sampling Technique	95.22	0.95	1	0.98	3403	170	1	3
2	Neural Network	SMOTE	80.42	0.95	1	0.98	3400	170	4	3
3	Neural Network	ADASYN	82.04	0.95	1	0.98	3404	169	0	4

Table 6: Neural Network Performance using Different Sampling Techniques

5.11 Sampling Methods Comparisons

According to the information presented in section 5.2, two distinct sampling strategies, namely SMOTE and ADASYN, were utilized in order to get a representative sample of the real data comprising 70 percent of the total. After that, the models that were generated with the data gathered by each of these methods were tested using the remaining thirty percent of the test data.
Based on the assessments and findings shown above, it can be concluded that SMOTE and ADASYN performed equally on Accuracy across the board (with the exception of the Neural Network model, where ADASYN performed somewhat better than SMOTE Similarly, techniques performed almost similarly for Random Forest (SMOTE Accuracy = 76.9, ADASYN Accuracy =77.3), Decision Tree (SMOTE Accuracy = 54.6, ADASYN Accuracy = 56.5),Adaboost(SMOTE Accuracy =55.8, ADASYN Accuracy=55.2), SVC(SMOTE Accuracy =50, ADASYN Accuracy=50)and Bagging (SMOTE Accuracy = 73, ADASYN Accuracy = 0.73), where ADASYN demonstrated a slight improvement over SMOTE. It is feasible to get the conclusion that ADASYN is the most appropriate sampling strategy for the task of stroke detection by taking into account the average accuracy of both of the sampling techniques.

Based on the data sampling methods used, the classification models built, and the findings obtained, it can be concluded that the Research Question and Sub-Research Question posed in section 1.2 were satisfactorily addressed, and that the project's objectives were met. Stroke detection is a challenging problem, and the ADASYN sampling technique was chosen as the best performing option. Based on the data sampling methods used, the classification models built, and the findings obtained, it can be concluded that the Research Question and Sub-Research Question posed in section 1.2 were satisfactorily addressed, and that the project's objectives were met. Stroke detection is a challenging problem, and the ADASYN sampling technique was chosen as the best performing option. It was found that this method was somewhat superior to SMOTE's over-sampling method when taken an average of both the methods. The study's findings will enrich both the current body of knowledge and the healthcare industry with the ultimate goal of identifying strokes at an earlier stage.

5.12 Discussion

According to the data that were obtained, this method appears to have been effective in addressing the issue of Stroke Detection. The most important contribution that may be made by implementing this solution is for the benefit of potential stroke patients as well as the whole healthcare sector. This technology has the potential to save lives and lessen the danger of lifelong impairments that are the result of strokes by identifying at an early-stage people who are at risk of having a stroke.
One of the most difficult parts of this project was figuring out how to deal with imbalanced datasets, and another was figuring out how to run and analyse 12 different machine learning model combinations and sampling techniques to figure out which model performed best and which models were most effective.
Numerous options were considered for dealing with the unbalanced dataset. To be honest, determining the most effective methods was a challenge. After a thorough search of the literature, the most widely used sampling strategies were uncovered. Two common sample approaches, each of a distinct type, were selected after an evaluation of several sampling procedures. After the examination, it was determined that ADASYN was the most effective method for advancing the Stroke Detection issue. Similar to the previous example, a total of

six alternative data analytics and machine learning classification models were selected for execution.

6 Conclusion and Future Work

Our project had the goal of producing a solution that would help to the advancement of the healthcare business by allowing for the early detection of stroke, which is one of the most significant life-threatening diseases of this era. With the assistance of data analytics, the aim was accomplished by putting into practice six distinct machine learning models, each of which utilized two distinct data sampling strategies. The Neural Network model, when combined with the ADASYN sampling technique, proved to be the most successful model because it was able to successfully differentiate between stroke cases and healthy cases, and it was also able to successfully identify 95 percent of the overall stroke cases (precision = 0.95) Neural Network model combined with the ADASYN sampling technique proved to be the best performing model because it was effectively capable of distinguishing between healthy cases and stroke cases.

Thereby, it can be said that this study has effectively answered the research question provided in Section 1.2, filling in all the gaps, and thus successfully addressing the problem of stroke detection.

Future Work:

The next step in this effort is to detect other disorders that pose a threat, such as trachea cancer, Ishemic heart disease, and diabetes. Further improvements can be made by identifying and implementing various special functions. To summarize, the goal of the future work is to create a strong toolset that can be used by all health providers and help anticipate all dangerous illnesses such tracheal cancer, ischaemic heart disease, and diabetes.

Acknowledgement

I'd want to take this opportunity to thank **Qurrat Ul Ain**, who serves as my supervisor. In the process of preparing the research work, her suggestions, which were both constructive and valuable, proved to be really helpful. I am extremely grateful to her for being ready to invest some of her time to assisting me during the process.

References

Tazin, T., Alam, M., Dola, N., Bari, M., Bourouis, S. and Monirujjaman Khan, M., 2021. Stroke Disease Detection and Prediction Using Robust Learning Approaches. *Journal of Healthcare Engineering*, 2021, pp.1-12

Rajora, M., Rathod, M. and Naik, N., 2021. Stroke Prediction Using Machine Learning in a Distributed Environment. *Distributed Computing and Internet Technology*, pp.238-252.

Sailasya, G. and Kumari, G., 2021. Analyzing the Performance of Stroke Prediction using ML Classification Algorithms. *International Journal of Advanced Computer Science and Applications*, 12(6).

Menezes, L., Gnanaraj, E., Bindra, S. and Pansare, A., 2021. Early-Stage Stroke Prediction Using Artificial Neural Network.

M. M. Islam, S. Akter, M. Rokunojjaman, J. H. Rony, A. Al Amin, and S. Kar, "Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique," Int. J. Electron. Commun. Syst., vol. 1, no. 2, 17-22, 2021.

Al-Islam, F. and Ghosh, M., 2021. An Enhanced Stroke Prediction Scheme Using SMOTE and Machine Learning Techniques.

Huang, X., Cao, T., Chen, L., Li, J., Tan, Z., Xu, B., Xu, R., Song, Y., Zhou, Z., Wang, Z., Wei, Y., Zhang, Y., Li, J., Huo, Y., Qin, X., Wu, Y., Wang, X., Wang, H., Cheng, X., Xu, X. and Liu, L., 2022. Novel Insights on Establishing Machine Learning Based Stroke Prediction Models Among Hypertensive Adults. *SSRN Electronic Journal*.

Wu, Y. and Fang, Y., 2020. Stroke Prediction with Machine Learning Methods among Older Chinese. *International Journal of Environmental Research and Public Health*, 17(6), p.1828.

Saleh, Hager & F Abd-el Ghany, Sara & Younis, Eman & Omran, Nahla & Ali, Abdelmgeid. (2019). Stroke Prediction using Distributed Machine Learning Based on Apache Spark. 10.13140/RG.2.2.13478.68162.

Chourib, I., Guillard, G., Farah, I. and Solaiman, B., 2022. Stroke Treatment Prediction Using Features Selection Methods and Machine Learning Classifiers. *IRBM*.

Kavitha, D., Jaisingh, D. and Sujithra, M., 2021. Applying Machine Learning Techniques for Stroke Prediction in Patients.

Zorkeflee, M., Din, A. and Mahamud, K., 2015. FUZZY AND SMOTE RESAMPLING TECHNIQUE FOR IMBALANCED DATA SETS.

Zeng, M., Zou, B., Wei, F., Liu, X. and Wang, L., 2016. Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data.

Alharbi, F., Ouarbya, L. and Ward, J., 2022. Comparing Sampling Strategies for Tackling Imbalanced Data in Human Activity Recognition. *Sensors*, 22(4), p.1373.

Rakshit, T. and Shrestha, A., 2021. Comparative Analysis and Implementation of Heart Stroke Prediction using Various Machine Learning Techniques.

Dev, S., Wang, H., Nwosu, C., Jain, N., Veeravalli, B. and John, D., 2022. A predictive analytics approach for stroke prediction using machine learning and neural networks. *Healthcare Analytics*, 2, p.100032.

JalajaJayalakshmi, M., Dr.V.Geetha, D. and Ijaz, M., 2021. Analysis and Prediction of Stroke using Machine Learning Algorithms.

GholamAzad, M., Pourmahmoud, J., Atashi, A., Farhoudi, M. and Deljavan Anvari, R., 2022. Predicting of Stroke Risk Based On Clinical Symptoms Using the Logistic Regression Method.

Hadianfard, Z., Lotfnezhad Afshar, H., Nazarbaghi, S., Rahimi, B. and Timpka, T., 2022. Predicting Mortality in Patients with Stroke Using Data Mining Techniques. *Acta Informatica Pragensia*, 11(1), pp.36-47.

Shafique, U. and Qaiser, H., 2014. A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA).