National College of Ireland

Identifying the Patients at Risk of Stroke Using Anomaly Detection Based Classification Approach

MSc Research Project
Data Analytics

Girish Jagwani
Student ID: x18136371

School of Computing
National College of Ireland

Supervisor: Dr. Catherine Mulwa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Girish Lakshman Jagwani
Student ID:	x18136371
Programme:	MSc. Data Analytics **Year:** 2019-20
Module:	MSc. Research Project
Supervisor:	Dr. Catherine Mulwa
Submission Due Date:	13-12-2019
Project Title:	Identifying the Patients at Risk of Stroke Using Anomaly Detection Based Classification Approach

Word Count: ..... **Page Count**.....

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: .....

Date: .....

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):	

Identifying the Patients at Risk of Stroke Using Anomaly Detection Based Classification Approach

Girish Jagwani
x18136371

Abstract

Stroke is one of the leading causes for death in 21st Century, accounting for the death of more than 2000 people in Ireland every year. Health Care Industry has done a lot of progress to cure stroke, but Stroke strikes suddenly, and the damage rate is so high that even if cured, it leaves permanent disabilities. The aim of this project is to identify the patients at risk of stroke using Electronic Health Records available with the hospitals and medical institutions. This is achieved by developing an Ensemble Voting Classifier with 9 different classification models as predictors. As the healthcare datasets are prone to be highly imbalanced, the 9 classification models along with the Ensemble Voting Classifier are developed and evaluated using 3 different sampling techniques. While evaluating the performance of all 30 modelled combinations, the combination of Ensemble Voting Classifier and hybrid sampling technique (SMOTE + Tomek) achieved the best results. The results obtained are promising and have successfully contributed towards the stroke detection problem in the healthcare industry.

1 Introduction

Healthcare Industry is one of the most crucial industries as it deals with lives. Healthcare industry has found the cure for many life-threatening diseases; However, even after cure, certain diseases can leave permanent disabilities. Also, such diseases need immediate medical attention due to their high damage rate to the human body and at times such diseases remain undetected because of absence or very rare presence of symptoms. One such disease is Stroke.

1.1 Background and Motivation

Stroke is medical emergency caused due to poor blood flow to the brain. Stroke accounted for the death of nearly 6 million people worldwide in 2016[1]. In Ireland alone, more than 2000 people die due to stroke every year, making it third most common cause of death[2]. The viciousness of stroke can be understood by the fact that it causes the death of more than two million brain cells every minute.

This technical report, therefore, focuses on detecting the patients at risk of having stroke using the power of machine learning. The work done so far to detect the patients at risk of stroke revolves mainly around the use of hardware instruments or predicting based on the medical examination reports like Magnetic Resonance Imaging (MRI). However, knowing that

[1] https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death
[2] https://www.independent.ie/regionals/sligochampion/news/stark-stroke-statistics-36508069.html

stroke strikes with rarely any prior symptoms[3], people considering themselves healthy will not undergo such examinations reducing the effectiveness of such approaches. Also, according to (McFadden, et al., 2009), social parameters too contribute towards stroke occurrence and not just physical health. This to the best of author's knowledge has never been taken into consideration while predicting stroke occurrence.

To cover this gap, this project focuses on using the electronic health records i.e., existing patient's data (both physical and social parameters) available with the Hospitals and Medical Institutions. This will not only avoid any overhead of medical examinations on the patients but will also improve the overall lifesaving rate.

1.2 Research Question

As in real-world, for the majority or almost all the hospitals, the number of patients at risk of stroke are expected to be far lesser than the total number of patients registered. Therefore, the dataset is always expected to be extremely imbalanced with minority class (patients at risk of having a stroke) being the key focus, which is a classic problem of Anomaly Detection based Classification. This leads to the following research question:

RQ: *"To what extent can the patients at risk of having stroke be identified by assessing their electronic health records using anomaly detection-based classification approach to therefore reducing the casualties caused due to stroke by taking precautionary measures?"*

The problem of imbalanced datasets can be addressed using sampling techniques and as per the literature reviewed, hybrid sampling techniques tend to provide better results when compared against Oversampling and Undersampling techniques. This leads to the following Sub-Research Question:

Sub RQ: *"To what extent can the hybrid sampling technique (SMOTE + Tomek) provide better results than Oversampling (SMOTE) and Undersampling (Tomek Links) for stroke detection?"*

The following section illustrates the objectives that were implemented to address the research questions.

1.3 Research Objectives

For successful implementation of any project, it is necessary to baseline a set of objectives that form its road map. Following are the objectives that are set as part of this project:

Objective 1: A critique on stroke detection and the sampling techniques used for the imbalanced datasets.

Objective 2: Data processing and feature selection for the detection of patients at risk of stroke.

Objective 3: Implementation, evaluation and results of stroke detection models using each of the 3 data sampling techniques (SMOTE, Tomek Links, SMOTE + Tomek).

[3] https://www.cdc.gov/stroke/

Objective 3.1: Implementation, evaluation and results of Extreme Gradient Boost (XGBoost).
Objective 3.2: Implementation, evaluation and results of Random Forest.
Objective 3.3: Implementation, evaluation and results of Support Vector Classifier (SVC).
Objective 3.4: Implementation, evaluation and results of Neural Network.
Objective 3.5: Implementation, evaluation and results of Naïve Bayes.
Objective 3.6: Implementation, evaluation and results of Logistic Regression.
Objective 3.7: Implementation, evaluation and results of K-Nearest Neighbours (KNN).
Objective 3.8: Implementation, evaluation and results of Decision Tree Classifier.
Objective 3.9: Implementation, evaluation and results of AdaBoost.
Objective 3.10: Implementation, evaluation and results of an Ensemble Voting Classifier using all 9 models developed above.
Objective 4: Comparison of all the developed models.

The key contribution of this research is to implement an ICT solution that will assist the hospitals and medical institutions to detect the patients that are at risk of having a stroke and therefore prescribe them with necessary preventive medication, reducing the overall stroke fatalities.

The rest of this technical report is structured as follows: Chapter 2 presents a critique on stroke detection and sampling techniques used for imbalanced datasets, Chapter 3 presents the scientific methodology approach and design for this project, Chapter 4 presents the implementation, evaluation and results of the stroke detection models, Chapter 5 presents a discussion followed by chapter 6 that presents the conclusion and suggested future work.

2 A Critique on Stroke Detection and Sampling Techniques Used for Imbalanced Datasets (2008-2019)

2.1 Introduction

This literature review investigates stroke detection and several techniques that are used to handle the imbalanced dataset. Section 2.2 investigates on the approaches that are used for stroke detection. Section 2.3 reviews on the classification models used in the case of highly imbalanced class distribution, for anomaly detection. Section 2.4 critiques on the techniques that are used to process imbalanced datasets. Followed by section 2.5 that narrates the identified gaps and section 2.6 that concludes this literature review.

2.2 An Investigation of the Approaches that are Used for Stroke Detection

A lot of work has been done in the recent time to identify the Stroke. Research work varies from the classification of stroke based on medical diagnosis to the classification of stroke based on facial movements. This section is intended to critique on the work done so far in this area.

2.2.1 A Review on Stroke Detection Using Image Processing and Monitoring Devices

Some common techniques for stroke detection mainly involve image processing or the use of devices such as wristbands to monitor the sleep rate as follows:

Researchers in (Chang, et al., 2018) and (Vijayalakshmi, et al., 2018) have detected stroke using the power of image processing. In (Chang, et al., 2018), researchers processed 69

different images of facial gestures by comparing facial gestures of the healthy individuals against the facial gestures of patients having a stroke. The machine learning models that were used for this process i.e., Naïve Bayes, Support Vector Machine and Random Forest achieved an accuracy of nearly 95% each. Whereas, in (Vijayalakshmi, et al., 2018), researchers performed similar research but on 32 images of Magnetic Resonance Imaging (MRI) of the patients using Support Vector Classifier and detected Stroke at an accuracy of 88%.

A different approach has been followed by researchers in (Jeon, et al., 2018) and (Xie, et al., 2018) where instead of using images, researchers have used hardware devices to capture the features that focus mainly on the sleeping habits of an individual to predict stroke occurrence. In (Jeon, et al., 2018), researchers used wrist bands to obtain parameters like sleep intensity and sleep frequency of 44 individuals (14 stroke patients and 30 healthy individuals) for the prediction. The Models that were used for this approach were Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) and on the relative scale of prediction, KNN was outperformed by SVM. A similar experiment conducted by (Xie, et al., 2018) using the sleep features of 225 individuals revealed that Support Vector Machine (SVM) performs better on the evaluation metrics of False Negative Value and True Positive Value when compared to models like Neural Network (NN), Random Forest and Naïve Bayes. Table 1 below summarises the findings based on the above researches.

Table 1 : Stroke Detection using Image Processing and Monitoring Devices

Authors	Pre-Requisite	Dataset Type	Number of Data Instances
(Chang, et al., 2018)	Facial Gesture Images	Image	69
(Vijayalakshmi, et al., 2018)	Magnetic Resonance Image	Image	32
(Jeon, et al., 2018)	Wrist Bands	Text	44
(Xie, et al., 2018)	polysomnogram	Text	225

The above approaches involve usage of image or device (to capture sleep-related features) turning them to be unscalable as it requires candidate's voluntary participation, which is unusual as stroke occurrence is considered as unpredictable with rarely any prior symptoms. Also, the volume data used for training and testing purposes is very confined questioning the reliability of the outcomes.

2.2.2 A Review on Stroke Detection Without Using Image Processing and Monitoring Devices

To tackle the problem of being unscalable due to candidate's voluntary participation for capturing images or wearing any devices, this section critiques the approaches that can be followed without any voluntary participation.

Researchers (Singh & Choudhary, 2017) have predicted stroke occurrence by assessing 357 different features of 1800 individuals (of which 200 were stroke patients). Feature selection and dimensionality reduction was done using Decision Tree and Principle Component Analysis respectively, which was then followed by the implementation of Artificial Neural Network

model that achieved an accuracy of 95% while predicting patients at risk of stroke. The downfall of this approach is that it is exhaustive to collect all the 357 features.

To predict stroke, researchers in (Jeena & Kumar, 2016) and (Sudha, et al., 2012) have assessed different psychological parameters and Gene Diagnostic Disease database respectively. While predicting stroke using psychological parameters of around 350 samples accuracy of 90% was achieved using Support Vector Machine (SVM) with linear kernel, Neural Network with an accuracy of 97% outperformed models like Decision Tree and Bayesian Classifier when applied on the 1000 entries of Gene Diagnostic Disease database.

In terms of predicting the stroke in the acute phase and its consequential risk (like eventual death), researchers in (Popukaylo, 2019) and (García-Terriza, et al., 2019) collected data of 250 and 120 individuals respectively. In (Popukaylo, 2019), Random Forest and XGBoost outperformed other models like Support Vector Machine (SVM), Decision Tree and Logistic Regression by achieving an accuracy of around 90%. Whereas, in the case of (García-Terriza, et al., 2019), while assessing on 6 different performance metrics, Random Forest outperformed all other models used (Support Vector Machine, Naïve Bayes, Logistic Regression, KNN, Decision Tree, Neural Network).

The above approaches have even though tackled the issue of being unscalable but due to the confined size of the dataset, overfitting always remains in question. Also, while predicting the stroke, these approaches do not account for the social parameters that may contribute to its occurrence (McFadden, et al., 2009).

2.3 An Investigation on the Classification Models Used for Anomaly Detection

As the real-world datasets are not always balanced, highly imbalanced datasets can lead the minority class to be treated as Anomaly. Machine learning models that are frequently used for the problem of stroke detection are already discussed as part of Section 2.2, this section discusses further on the Machine learning models that are used for anomaly detection problems across different domains.

To detect microcalcification clusters that are key in the earlier identification of breast cancer, researchers in (Ren, 2012) assessed mammography images using models like Support Vector Machine (SVM) and Artificial Neural Network (ANN) and on a relative scale of F1 score, ANN showed an improvement of 10% over SVM.

To spot faults in the cloud infrastructure, researchers in (Gulenko, et al., 2016) assessed a highly imbalanced dataset treating faults as anomalies that were injected using offline fault injection experiment. Results suggest that models like Support Vector Machine (SVM), Naïve Bayes, Decision Tree, Logistic Regression and Random Forest predicted anomalies with an average F1 – score of 91%. Similarly, while forecasting monsoon, Researchers (Troncoso, et al., 2018) evaluated classification models like Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors, Decision Tree and Artificial Neural Network on the performance metrics of Recall and Positive Predictive Value and Random Forest turned out to provide a relatively balanced score.

In (Oughali, et al., 2019), researchers used models like XGBoost and Random Forest to analyse over 2,00,000 shots of NBA players for the season of 2014-2015 and predicted the shots at an accuracy of 68% and 57% respectively.

2.4 A Critical Review on Techniques to Process Imbalanced Datasets

Sampling is the most popular technique to process the imbalanced datasets for achieving class balance (Guo, et al., 2008). However, sampling is a broader term as there are different types of samplings like over-sampling and under-sampling. Also, there are different methods for both over-sampling and under-sampling. This section, therefore, aims at critically reviewing on the different sampling techniques that can be used for balancing the datasets.

To tackle the problem of class imbalance while detecting intrusions, author (Qazi & Raza, 2012) assessed the effects of both under-sampling and over-sampling. The outcome suggests that while sampling is an effective approach towards balancing the dataset, under-sampling provides better results when the goal is to identify the minority class as compared to over-sampling.

While working with highly imbalanced datasets for classifying Autism and predicting Monsoon, researchers in (El-Sayed, et al., 2015) and (Troncoso, et al., 2018) have done oversampling of the minority class using Synthetic Minority Over-sampling Technique (SMOTE). In (El-Sayed, et al., 2015), researchers suggest that although over-sampling is vulnerable to overfitting, the accuracy obtained for models like Support Vector Machine (SVM), Naïve Bayes and Decision Tree is certainly less deceptive. Similarly, in (Troncoso, et al., 2018) researchers while evaluating models like Neural Network, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Decision Tree Classifier on performance metrics of Positive Predictive Value and Recall suggest that oversampling using SMOTE has tackled the problem of class imbalance successfully.

Another important type of sampling is the hybrid or combined sampling that involves over-sampling of the minority class followed by the under-sampling of the majority class.

To evaluate the effectiveness of different sampling techniques, researchers in (Batista, et al., 2004) have sampled 13 datasets. While evaluating the results using performance metrics of Area under the ROC curve (AUROC), the author suggests that hybrid sampling of SMOTE + ENN and SMOTE + Tomek achieved relatively better results for skewed datasets when compared against under-sampling and over-sampling alone. Similar results were obtained when researchers in (Sain & Purnami, 2015) evaluated the performance of SMOTE + Tomek against SMOTE and Tomek Links using Support Vector Machine (SVM) model on an imbalanced dataset of healthcare in 5-fold cross-validation.

Similarly, in the healthcare industry where imbalanced datasets are relatively more common, while predicting certain types of diseases or if an individual is healthy or has the disease, researchers (Zeng, et al., 2016) and (Elhassan & Aljurf, 2017) have tried using different sampling techniques for handling imbalanced datasets. In (Zeng, et al., 2016), to predict diseases like Parkinson's disease or diabetes, researchers have used hybrid sampling (SMOTE + Tomek Links), Over-sampling (SMOTE) and Under-sampling (Tomek-Links) and the evaluation suggest that Hybrid sampling tends to provide better results. Whereas, researchers in (Elhassan & Aljurf, 2017), while analysing the EColi2 and arterial blood pressure-related data to identify if an individual is healthy or has disease using machine learning models like Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest and Logistic Regression conclude that SMOTE and RUS provides best results when combined with Tomek Links.

Based on the above discussion, it can be concluded that even though Hybrid sampling usually tends to provide more balanced outcomes than Over-sampling and Under-sampling, it cannot guarantee better results and the effectiveness of sampling technique is completely dataset-specific.

2.5 Identified Gaps

Based on the Critique of the approaches that are used for Stroke Identification in section 2.2, below are the gaps that are identified with the best of the candidate's knowledge and covered as part of this project.

Overall the Stroke detection approaches can be divided into 2 parts:

The first approach, as reviewed in section 2.2.1, involves stroke detection using image processing and hardware devices. This approach has a gap as it is not scalable as it requires candidate's voluntary participation, which is rare as Stroke occurrence is considered as unpredictable with very rarely any symptoms[4]. Also, the dataset size is very confined making the machine learning models vulnerable to overfitting.

The second approach, as reviewed in section 2.2.2, involves stroke detection using health records. However, for all such related work, size of the dataset that is used is very confined. This makes the machine learning models vulnerable to overfitting. Also, according to (McFadden, et al., 2009), social parameters too can contribute to the occurrence of stroke which is a gap as it has never been taken into consideration before.

2.6 Conclusion

Based on the critique done as part of this chapter, there are several identified gaps as discussed in section 2.5. To cover this gap, as part of this project, Electronic Health Record dataset is used for stroke detection that consists of more than 43000 health records, reducing the vulnerability of models towards overfitting. As the dataset has highly imbalanced class distribution, 3 different sampling techniques are used, i.e., SMOTE (over-sampling), Tomek Links (under-sampling) and SMOTE + Tomek (hybrid sampling) for the implementation of the classification models like eXtreme Gradient Boosting (XGBoost), Random Forest, Neural Network, Support Vector Classifier (SVC), Decision Tree Classifier, AdaBoost Classifier, Logistic Regression, K-Nearest Neighbors (KNN), and Naïve Bayes. These models are then ensembled into a Voting Classifier for soft voting. The evaluation is performed based on the performance metrics of Confusion Matrix, Specificity, Sensitivity, Area under the ROC curve (AUROC) and Accuracy. As the dataset is imbalanced and use Accuracy is not recommended, it is only used as a supplementary evaluation metrics. With this critique on stroke detection, Objective 1 of section 1.3 is now achieved.

[4] https://www.cdc.gov/stroke/

3 Scientific Methodology Approach and Design

For the development of any data analytics project, there are 3 widely used methodologies,i.e., Sample Explore Modify Model and Access (SEMMA), Knowledge Discovery and Data mining (KDD) and CRoss Industry Standard Process for Data Mining (CRISP-DM). While SEMMA and KDD are quite equivalent, CRISP-DM is more Business focused, (Azevedo & Santos, 2008). Considering the aim of this project is to detect the patients that are at risk of having a stroke and that the business layer deployment is not in the scope of this project, the decision was taken to choose Modified KDD (Fayyad, et al., 1996), over CRISP-DM and SEMMA.

3.1.1 Modified Knowledge Discovery and Data mining (KDD) Approach

Knowledge Discovery and Data Mining (KDD) approach, as its name suggests is more focused on the data mining when compared to CRISP-DM which is more business-focused. After choosing KDD approach, it was then modified further to fit it in the context of this project. Figure 1 demonstrates the modified KDD approach that is used as part of this project.

KDD consist of 5 different stages that sums up the implementation of this entire project, those stages in the context of stroke detection are as follows:

1. **Data Selection**
   This stage consists of the process of selecting and extracting the data from respective data sources. In this case, Electronic Health Records dataset is extracted from Kaggle[5]. The outcome of this stage is the selected data that serves as an input for the data pre-processing stage.
2. **Data Pre-processing**
   Data Pre-processing stage includes activities like exploratory data analysis along with data imputation and feature selection as illustrated in section 4.2. The outcome of this stage is the pre-processed data that serves as an input for the data transformation stage.
3. **Data Transformation**
   Data Transformation activity that is undertaken as part of this project includes standardisation of data using sklearn.preprocessing.StandardScaler library of python. The outcome of this stage is the transformed data that serves as an input for the data mining stage.
4. **Data Mining**
   This is the most crucial stage of the entire approach as it involves implementation of the data mining techniques to identify patterns in the data. As part of this project, 9 different classification models are developed using 3 different data sampling techniques each, which are then ensembled into the respective Ensemble Voting Classifiers based on data sampling technique used. The outcome of this stage is the patterns identified into the data that is therefore evaluated as part of the evaluation stage.
5. **Interpretation/Evaluation**
   The models trained as part of data mining stage and the respective patterns identified are evaluated as part of this stage based on performance metrics of the confusion matrix, sensitivity, specificity, AUROC and Accuracy. The outcome of this final stage of KDD

[5] https://www.kaggle.com/asaumya/patient-data-train-and-test-set/metadata

is the knowledge that is gained and used for future predictions of stroke. This completes the entire lifecycle of KDD methodology, the same is illustrated in Figure 1.

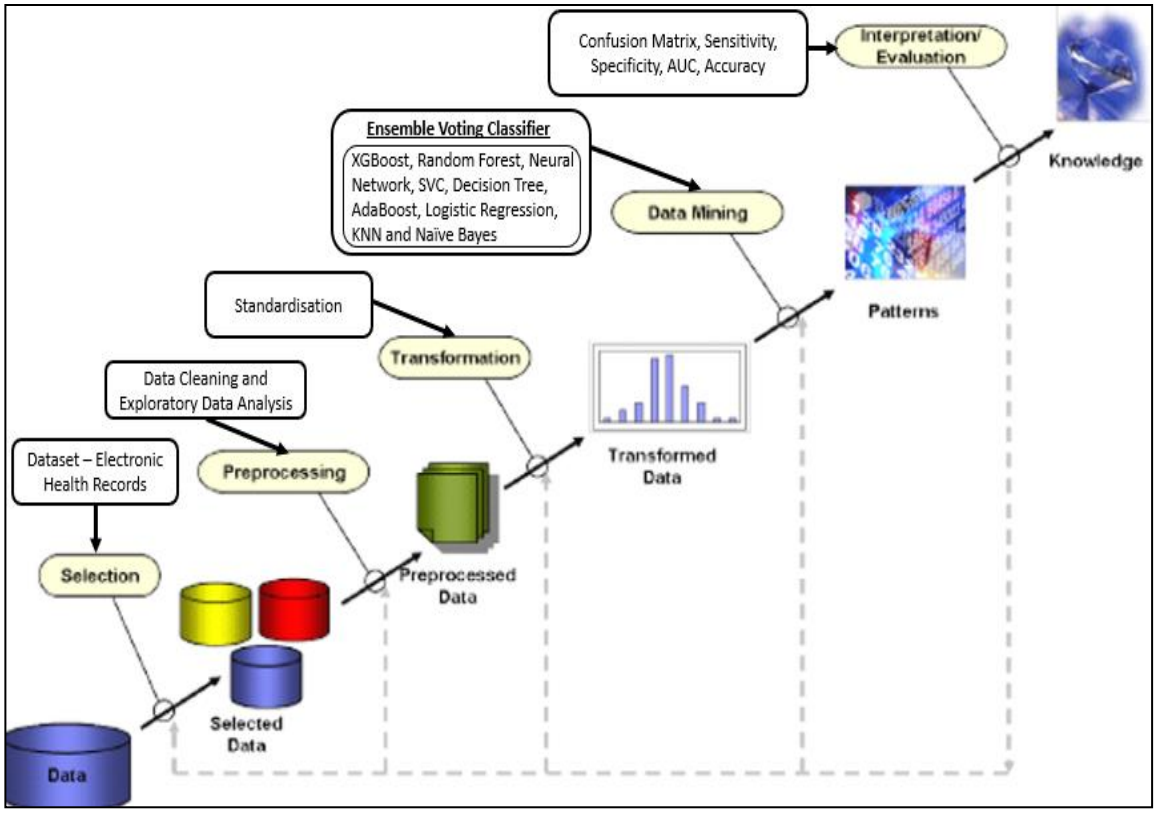

Figure 1 : Modified KDD Approach for Stroke Detection

3.2 Design Specification

The project design specifications summarise the overall architecture of the project with the detail like the process flow, tools, technologies and the techniques to be used for the implementation of this project. It takes all the implementation and evaluation objectives that are specified under section 1.3 into consideration.

For any data analytics project, the architectural design can be classified as either 2-Tier design or 3-Tier design. This project of stroke detection follows the 3-Tier design as illustrated in Figure 2.

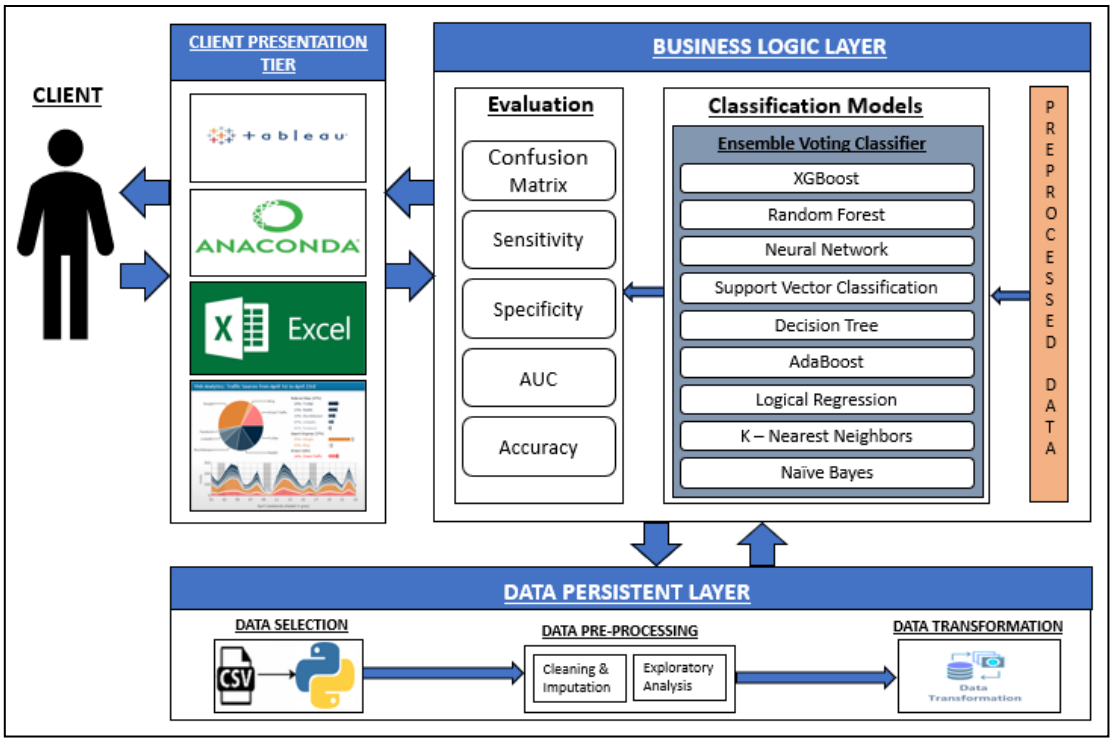

Figure 2 : 3-Tier project design for Stroke Detection

The three tiers of this project are Client Presentation Layer, Business Logic Layer and Data Persistent Layer. All the tiers are inter-connected with each other for to facilitate the flow of the data.

1. **Data Persistent Layer**
   As its name suggests, this layer deals directly with the raw data and process it in Python by performing activities like data cleaning, imputing the missing values, transforming the data using standardising techniques and passes the processed data to the Business Logic Layer.
2. **Business Logic Layer**
   This layer is the actual implementation layer where all business logics are implemented. It receives the processed data from the data persistent layer and applies 9 different classification models that are then ensembled into a voting classifier for soft voting. This model is then evaluated based on the performance metrics of Confusion Matrix, Sensitivity, Specificity, Area Under the ROC Curve (AUC) and Accuracy.
3. **Client Tier / Presentation Layer**
   This is the only visible layer to the client and is responsible for taking inputs from the client and providing the respective output of business logic layer to the client. The output is in terms of data using Microsoft Excel and Visualisations using Anaconda (Python) and Tableau.

To conclude, considering the aim of this project and all the objectives specified in section 1.3, the combination of modified KDD and 3-Tier Design was chosen to follow a more data-centric approach.

4 Implementation, Evaluation and Results of the Stroke Detection Models

4.1 Introduction

This chapter discusses on the implementation of the 10 different models that were developed using 3 different sampling techniques each, along with their evaluation and results. It also discusses on the preliminary data processing steps like Exploratory Data Analysis (EDA), Data Cleansing and Processing, Feature Selection, Data Scaling that were undertaken prior to the implementation.

This introduction further illustrates the tools along with the sampling techniques and evaluation metrics used for this project.

4.1.1 Tools Used for Implementation

The implementation of this project along with the visualisation of the data was done using Anaconda Python and the scripts were written in Spyder IDE. Microsoft Excel is used to facilitate the users to view the 4 .csv files generated as output, that consist 3 .csv files of the predictions done by the model using different sampling techniques and a .csv file with the evaluation metrics for all the implemented models.

4.1.2 Sampling Techniques

On completion of the activities like data processing and feature selection, the dataset was divided into Training and Testing data in a stratified ratio of 70:30 using train_test_split() function of sklearn.model_selection package.

Based on the review in section 2.4, 3 different sampling techniques were chosen to be applied on the training data. Those are as follows:

1. **Synthetic Minority Over-sampling Technique (SMOTE)**
   SMOTE is an over-sampling technique that is used to generate synthetic instances of the minority class to achieve the class balance. It was implemented on the training dataset using SMOTE() function of imblearn.over_sampling package in Python.
2. **Tomek Links**
   Tomek Links is an under-sampling technique that removes the border cases from the dataset. It was implemented on the training dataset using TomekLinks() function of imblearn.under_sampling package in Python.
3. **SMOTE + Tomek**
   SMOTE + Tomek is a hybrid sampling approach that consists of the implementation of both SMOTE and Tomek Links. SMOTE generates synthetic samples to attain class balance. However, SMOTE does not take border cases into consideration. To mitigate this issue, Tomek Links is implemented after SMOTE. It was implemented using SMOTETomek() function of imblearn.combine package in Python.

4.1.3 Evaluation Metrics for Stroke Detection Models

The evaluation was performed using the 30% test data that was created during data sampling activities, by dividing the actual dataset in a stratified ratio of 70:30 for training and testing respectively. The performance metrics that are used for the evaluation as part of this project

are Confusion Matrix, Specificity, Sensitivity and Area Under the ROC Curve. Due to the class imbalance, the use of Accuracy as a performance metrics is deprecated. Therefore, it is only used as supplementary to the other metrics and not for actual evaluations.

1. **Confusion Matrix**

   Confusion Matrix helps in summarising the outcome of the predictions in terms of a matrix consisting of True Positives (correctly predicted stroke cases), True Negative (correctly predicted healthy cases), False Positive (healthy cases incorrectly predicted as stroke) and False Negative (stroke cases incorrectly predicted as healthy). Confusion matrix forms the foundation of all the performance metrics. Figure 3 shows the confusion matrix to be used for this project.

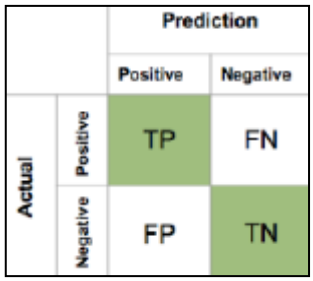

Figure 3: Confusion Matrix

2. **Specificity (True Negative Rate)**

   It is the proportion of correctly predicted healthy cases against the actual number of healthy cases. It is calculated using the following formula:

$$Specificity = \frac{(TN)}{(TN + FP)}$$

3. **Sensitivity (Recall / True Positive Rate)**

   It is the proportion of correctly predicted stroke cases against the actual number of stroke cases. It is calculated using the following formula:

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

4. **Area Under the ROC Curve (AUROC)**

   It is the ability of the models to correctly distinguish between healthy cases and stroke cases. It ranges from 0 to 1 with 1 indicating an ideal model with 100% distinguishing ability.

5. **Accuracy**

   It is the proportion of correctly predicted stroke and healthy cases against the total number of predictions. It is calculated using the following formula:

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

4.2 Exploratory Data Analysis, Data Pre-processing and Feature Selection

This section focuses on all the processing steps that were taken prior to the implementation of Machine Learning Models.

4.2.1 Dataset Overview

The dataset used as part of this project is an Electronic Health Record dataset that is publicly available on Kaggle[6]. As Healthcare industry datasets are prone to the problem of

[6] https://www.kaggle.com/asaumya/patient-data-train-and-test-set/metadata

imbalanced class distribution, the class distribution of this dataset is also highly imbalanced with the minority class of stroke cases accounting for only 783 records out of the total of 43,400.

This dataset is GDPR and Ethics compliant as the information of the participants is completely anonymised by the source.

4.2.2 Exploratory Data Analysis

Exploratory Data Analysis is essential for understanding the Electronic Health Record dataset in a better way. All the variables, their class balance or distribution along with the relationship in association to each other were analysed as part of this stage. As this stage consists of a lot of analysis and visualisations which is not feasible to showcase in this document (due to the size constraints), this document highlights only on some of the important analysis done and the further analysis and visualisations are covered as part of the configuration manual.

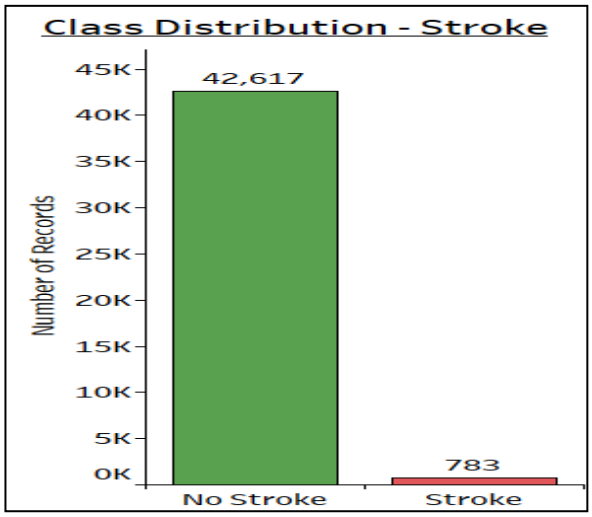

Figure 4 : Stroke Class Distribution

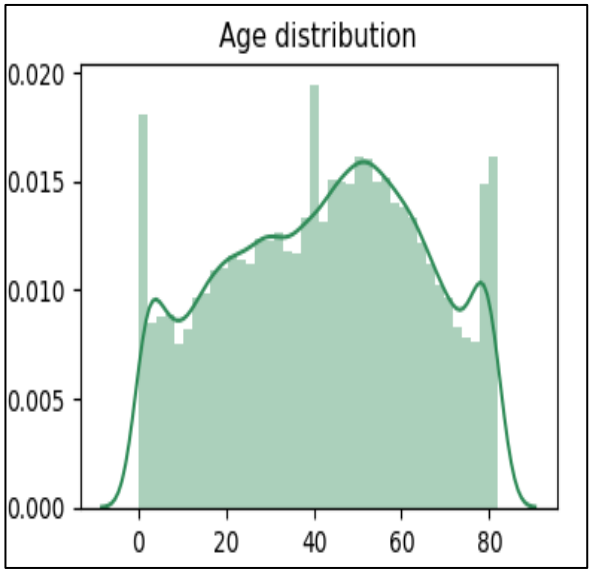

Figure 5 : Age Distribution

While trying to analyse the class distribution of the dependent variable (Stroke), as illustrated in the visualisation in Figure 4 it was identified that the dataset is highly imbalanced with only 783 cases for stroke reported against the 42,617 instances of healthy cases (in context of stroke).

A similar analysis was performed for other categorical variables like Gender, Hypertension, Heart Disease, Work Type, Residence Type and Smoking Status.

In case of continuous variables like Age, Body Mass Index (BMI), Average Glucose level a distribution plot was created using distplot() function of seaborn package in python to analyse the data further.

Figure 5 illustrates that the distribution of age is quite consistent across the dataset. However, while evaluating the relationship between age, gender and stroke using a violin plot in Python, Figure 6 suggest that stroke is more likely to occur at an elderly age of above 40 when compared against the age of 40 and below. Also, Females are more likely to get a stroke in the younger age of 20 and below and in the elder age of nearly 80 when compared against males.

Figure 6 : Gender vs Age vs Stroke

```
************ bmi ************
False    41938
True      1462
Name: bmi, dtype: int64
****************************************

************ smoking_status ************
False    30108
True     13292
Name: smoking_status, dtype: int64
****************************************
```

Figure 7 : Missing Values in the Dataset

Apart from identifying relations and evaluating the data quality, another important purpose of Exploratory Data Analysis was to identify the inconsistent and Missing Values in the dataset. While investigating for the missing values using python it was identified that as illustrated in Figure 7 there were 1462 missing values in BMI, whereas 13292 missing values in Smoking_Status.

4.2.3 Data Processing and Feature Selection

Data Processing and Feature Selection help to process the raw data and get the relevant features out of it, making it suitable for the implementation by Machine Learning models. The major activities that were done as part of this stage are as follows:

1. **Imputation of Null Values**

   Based on the Exploratory Data Analysis is done, only 1,462 Null values were present in bmi column and therefore those were imputed by Multivariate Imputation by Chained Equation (MICE) technique using mice() function of impyute.imputation.cs package in Python.

   In case of smoking_status, 13,292 Null values were identified which account for almost 33% of the total records because of which imputation was not recommended. Therefore, the missing values were replaced by referring the age column. Since minimum age for smoking is 18 years, smoking_status for respective Null records was updated as “Never Smoked”. For the records with age above 18 years and smoking_status as null, smoking status was updated as “unknown”.

2. **Encoding of Categorical Variables**

   To make the machine learning models understand categorical variables, it needs to be encoded into integers. Encoding was performed for 7 categorical variables in Python using LabelEncoder() function of sklearn.preprocessing package.

3. **Feature Selection and Data Scaling**

   Feature Selection is important to provide only relevant features to the machine learning models to improve performance and avoid the model fitting issues. Based on the data understanding, features like id along with the other redundant prior-encoding categorical features were removed. Also, Correlation among the features was identified to ensure the absence of multicollinearity which is one of the basic assumptions for models like Logistic Regression. Based on the visualisation in Figure 8, at a threshold of 0.7, it can be concluded that multicollinearity does not exist in the selected features.

   Data Scaling of the continuous variables was done in Python using StandardScalar() function of sklearn package.

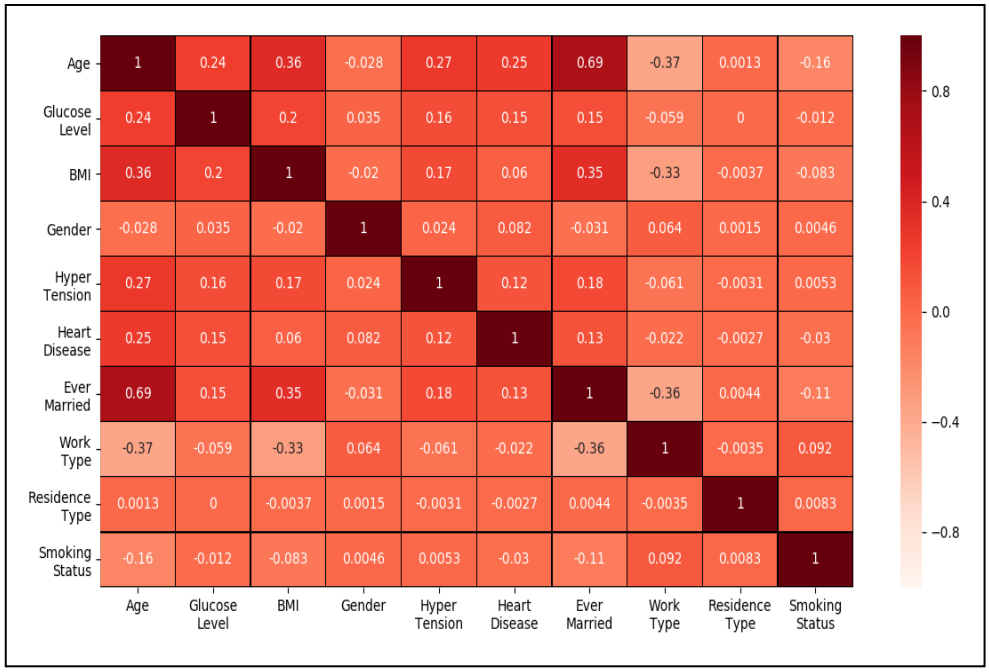

Figure 8 : Correlation Plot for Stroke Detection

.

4.3 Implementation, Evaluation and Results of eXtreme Gradient Boost (XGBoost)

XGBoost is an Ensemble machine learning model that is built over the framework of Gradient Boosting which uses several weak learners to develop a strong machine learning model. The advantage of using XGBoost is that it makes optimum use of the available resources and implements parallel processing to predict the outcomes.

Implementation:

XGBoost was implemented for each of the 3 sampling techniques using XGBClassifier() function of xgboost package. On tuning the parameters, the best results were achieved with the value of max_depth as 4 and the weights assigned to scale_post_weight parameter as the square

root of the class weights calculated using compute_class_weight() function of sklearn.utils. class_weight package. Objective parameter was set as "binary:logistic" as stroke detection is a problem of binary classification.

Evaluation and Results:

Table 2 illustrates the performance of the 3 different sampling techniques when used with XGBoost. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is evident that Tomek Links *(AUROC = 0.52)* performed the worst. Whereas, performance was similar for SMOTE *(AUROC = 0.73)* and SMOTE + Tomek *(AUROC = 0.73)*. However, the relatively higher specificity and True Negative value for SMOTE + Tomek *(Specificity = 0.69, TN = 5851)* suggest that it identified the healthy cases slightly better than SMOTE *(Specificity = 0.68, TN = 5812)*. Therefore, it can be concluded that XGBoost performed best with SMOTE + Tomek sampling technique.

Table 2 : Performance of XGBoost with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	XGBoost	SMOTE	0.78	0.68	0.73	0.68	122	35	2711	5812
2	XGBoost	TOMEKLINKS	0.04	1	0.52	0.98	6	151	24	8499
3	XGBoost	SMOTETOMEK	0.78	0.69	0.73	0.69	122	35	2672	5851

4.4 Implementation, Evaluation and Results of Random Forest

Random Forest is an ensemble of decision trees where each decision tree is assigned with a random set of features and random data sample. The mode of the outcome of all these trees is the outcome of Random Forest. The advantage of using Random Forest is that it tends to provide lower error rate and is less prone to overfitting.

Implementation:

Random Forest was implemented for each of the 3 sampling techniques using RandomForestClassifier() function of sklearn.ensemble package in Python. On tuning the parameters, the optimum results were achieved with n_estimator as 100 and max_depth as 2.

Evaluation and Results:

Table 3 illustrates the performance of the 3 different sampling techniques when used with Random Forest. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is can be observed that all the 3 sampling techniques have achieved similar performance *(AUROC = 0.78)*. However, SMOTE *(Sensitivity = 0.83, Specificity = 0.72)* and SMOTE + Tomek *(Sensitivity = 0.83, Specificity = 0.72)* provided better balance between Sensitivity and Specificity when compared against Tomeklinks *(Sensitivity = 0.91, Specificity = 0.66)*. Therefore, it can be concluded that Random Forest provided better-balanced results with SMOTE and SMOTE + Tomek when compared against Tomek Links.

Table 3: Performance of Random Forest with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	Random Forest	SMOTE	0.83	0.72	0.78	0.72	131	26	2361	6162

2	Random Forest	TOMEKLINKS	0.91	0.66	0.78	0.66	143	14	2938	5585
3	Random Forest	SMOTETOMEK	0.83	0.72	0.78	0.72	131	26	2383	6140

4.5 Implementation, Evaluation and Results of Support Vector Classifier (SVC)

Support Vector Classifier (SVC) is a model that uses the hyperplane to distinguish between the outcomes of a classification.

Implementation:

SVC was implemented with sigmoid kernel for each of the 3 sampling techniques using SVC() function of sklearn.svm package in Python. On tuning the parameters, optimum results were achieved with max_iter as 12000, C as 25 and gamma as 0.03. The value for probability parameter for SVC was set to True, to make it usable for Ensemble Voting Classifier that was implemented later.

Evaluation and Results:

Table 4 illustrates the performance of the 3 different sampling techniques when used with Support Vector Classifier. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is evident that Tomek Links *(AUROC = 0.52)* performed the worst. Whereas, performance was similar for SMOTE *(AUROC = 0.70)* and SMOTE + Tomek *(AUROC = 0.70).* Therefore, it can be concluded that Support Vector Classifier provided better results with both SMOTE and SMOTE + Tomek when compared against Tomek Links.

Table 4 : Performance of Support Vector Classifier with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	SVC	SMOTE	0.68	0.70	0.69	0.70	107	50	2594	5929
2	SVC	TOMEKLINKS	0.06	0.98	0.52	0.97	10	147	137	8386
3	SVC	SMOTETOMEK	0.69	0.70	0.69	0.70	108	49	2590	5933

4.6 Implementation, Evaluation and Results of Neural Network

Neural Network model consists of neurons across several layers that understand the behaviour of the data to identify the underlying pattern which is then used for predictions.

Implementation:

Neural Network was implemented for each of the 3 sampling techniques using KerasClassifier() function of keras.wrappers.scikit_learn package in Python. On tuning the parameters, optimum results were achieved with epochs as 1000, batch_size as 2000, validation split as 0.33 and class_weight in accordance to the value identified using compute_class_weight() function of sklearn.utils.class_weight package.

Evaluation and Results:

Table 5 illustrates the performance of the 3 different sampling techniques when used with Neural Network. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is evident that SMOTE + Tomek *(AUROC = 0.71)* performed best when compared against SMOTE *(AUROC = 0.69)*

and Tomek Links *(AUROC = 0.69)*. Also, Sensitivity and Specificity that provides the proportion of correctly predicted stroke and health cases respectively is highest for SMOTE + Tomek *(Sensitivity = 0.68, Specificity = 0.74)*. Therefore, it can be concluded that Neural Network performed best with SMOTE + Tomek sampling technique.

Table 5 : Performance of Neural Network with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	Neural Network	SMOTE	0.66	0.73	0.69	0.73	103	54	2282	6241
2	Neural Network	TOMEKLINKS	0.58	0.80	0.69	0.79	91	66	1746	6777
3	Neural Network	SMOTETOMEK	0.68	0.74	0.71	0.74	106	51	2188	6335

4.7 Implementation, Evaluation and Results of Naïve Bayes

Naïve Bayes classifier predicts the outcomes of the dependent variable based on probabilities.

Implementation:

Gaussian Naïve Bayes was implemented for each of the 3 sampling techniques using GaussianNB() function of sklearn.naive_bayes package in Python.

Evaluation and Results:

Table 6 illustrates the performance of the 3 different sampling techniques when used with Naïve Bayes. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is evident that Tomek Links *(AUROC = 0.66)* performed the worst. Whereas, performance was similar for SMOTE *(AUROC = 0.77)* and SMOTE + Tomek *(AUROC = 0.77)*. Therefore, it can be concluded that Naïve Bayes provided better results with both SMOTE and SMOTE + Tomek when compared against Tomek Links.

Table 6 : Performance of Naive Bayes with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	Naive Bayes	SMOTE	0.85	0.69	0.77	0.70	134	23	2611	5912
2	Naive Bayes	TOMEKLINKS	0.40	0.92	0.66	0.91	63	94	712	7811
3	Naive Bayes	SMOTETOMEK	0.85	0.79	0.77	0.70	134	23	2612	5911

4.8 Implementation, Evaluation and Results of Logistic Regression

Logistic Regression model identifies the relationship between the predictors and the dependent variable to predict the likelihood of the outcome.

Implementation:

Logistic Regression model was implemented for each of the 3 sampling techniques using LogisticRegression() function of sklearn.linear_model package in python. On tuning the parameters, optimum results were achieved with class_weight as balanced and penalty as l2.

Evaluation and Results:

Table 7 illustrates the performance of the 3 different sampling techniques when used with Logistic Regression. While evaluating the performance against the metrics like Sensitivity,

Specificity, AUROC and Accuracy, it can be observed that all the three techniques have achieved similar results *(Sensitivity = 0.85, Specificity = 0.73, AUROC = 0.79, Accuracy = 0.73*). Therefore, it can be concluded that Logistic Regression performed independent of the sampling technique type used.

Table 7 : Performance of Logistic Regression with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	Logistic Regression	SMOTE	0.85	0.73	0.79	0.73	133	24	2300	6223
2	Logistic Regression	TOMEKLINKS	0.85	0.73	0.79	0.73	133	24	2314	6209
3	Logistic Regression	SMOTETOMEK	0.85	0.73	0.79	0.73	134	23	2301	6222

4.9 Implementation, Evaluation and Results of K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) model predicts the class of a given test sample by observing the class of the majority its nearest neighbours, without undertaking any prior assumptions.

Implementation:

KNN was implemented for each of the 3 sampling techniques using KNeighborsClassifier() function of sklearn.neighbors in Python. On tuning the parameters, optimum results were achieved with the value for K, i.e., n_neighbors as 70.

Evaluation and Results:

Table 8 illustrates the performance of the 3 different sampling techniques when used with KNN. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is evident that Tomek Links *(AUROC = 0.5)* performed the worst. Whereas, performance was similar for SMOTE *(AUROC = 0.72)* and SMOTE + Tomek *(AUROC = 0.72).* Therefore, it can be concluded that KNN provided better results with both SMOTE and SMOTE + Tomek when compared against Tomek Links.

Table 8 : Performance of K-Nearest Neighbors with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	KNN	SMOTE	0.71	0.72	0.72	0.72	112	45	2356	6167
2	KNN	TOMEKLINKS	0	1	0.5	0.98	0	157	0	8523
3	KNN	SMOTETOMEK	0.71	0.72	0.72	0.72	112	45	2356	6167

4.10 Implementation, Evaluation and Results of Decision Tree Classifier

Decision Tree classifier model is an iterative cycle where a condition is evaluated as a node and its outcomes are split into several branches, the process continues based on the parameters set until it concludes with leaves that predict the outcome of the stroke detection.

Implementation:

Decision Tree was implemented for each of the 3 sampling techniques using DecisionTreeClassifier() function of sklearn.tree package in Python. On tuning the parameters,

optimum results were achieved with max_depth set to 10, criterion set to ‘entropy’ and weight set to ‘balanced’.

Evaluation and Results:

Table 9 illustrates the performance of the 3 different sampling techniques when used with Decision Tree Classifier. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is can be observed that all the 3 sampling techniques have achieved similar performance *(AUROC = 0.71 to 0.73).* However, SMOTE and SMOTE + Tomek (*Sensitivity = 0.70, Specificity = 0.75)* provided better balance between Sensitivity and Specificity when compared against Tomeklinks *(Sensitivity = 0.62, Specificity = 0.79).* Therefore, it can be concluded that Decision Tree Classifier provided better-balanced results with SMOTE and SMOTE + Tomek when compared against Tomek Links.

Table 9 : Performance of Decision Tree with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	Decision Tree	SMOTE	0.70	0.75	0.73	0.75	110	47	2125	6398
2	Decision Tree	TOMEKLINKS	0.62	0.79	0.71	0.79	98	59	1752	6771
3	Decision Tree	SMOTETOMEK	0.70	0.75	0.73	0.75	110	47	2126	6397

4.11 Implementation, Evaluation and Results of AdaBoost (Adaptive Boost)

AdaBoost is an Ensemble Boosting model that forms a strong classifier by combining numerous weak classifiers. The best performing models are then given higher weights in AdaBoost to improve overall outcomes.

Implementation:

AdaBoost classifier was implemented for each of the 3 sampling techniques using AdaBoostClassifier() function of sklearn.ensemble package in Python. On tuning the parameters, optimum results were achieved with n_estimators as 5 and learning_rate as 1.0.

Evaluation and Results:

Table 10 illustrates the performance of the 3 different sampling techniques when used with AdaBoost. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is evident that Tomek Links *(AUROC = 0.50)* performed the worst. Whereas, performance was similar for SMOTE *(AUROC = 0.77)* and SMOTE + Tomek *(AUROC = 0.77).* Therefore, it can be concluded that AdaBoost provided better results with both SMOTE and SMOTE + Tomek when compared against Tomek Links.

Table 10 : Performance of AdaBoost with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	AdaBoost	SMOTE	0.82	0.72	0.77	0.72	129	28	2414	6109
2	AdaBoost	TOMEKLINKS	0	1	0.5	0.98	0	157	0	8523
3	AdaBoost	SMOTETOMEK	0.82	0.72	0.77	0.72	129	28	2414	6109

4.12 Implementation, Evaluation and Results of Ensemble Voting Classifier

An Ensemble Voting Classifier was created using all the 9 classification models implemented above as predictors.

An Ensemble Voting Classifier is a model that uses several other models as predictors. These models then vote for the probability of the outcome and the outcome with the highest probability is predicted as the outcome of Ensemble Voting Classifier.

Implementation:

Ensemble Voting Classifier was implemented for each of the 3 sampling techniques using VotingClassifier() method of sklearn.ensemble package in Python. All the 9 tuned machine learning models were passed as estimators for this classifier and the value for 'voting' parameter was set to 'soft'.

Evaluation and Results:

Table 11 illustrates the performance of the 3 different sampling techniques when used with Ensemble Voting Classifier. While evaluating the performance against AUROC metrics that illustrates the ability of the model to distinguish between stroke and healthy cases, it is evident that Tomek Links *(AUROC = 0.56)* performed the worst. Whereas, the performance was similar for SMOTE and SMOTE + Tomek *(AUROC = 0.79)*. However, when compared the performance against Sensitivity, Specificity and Accuracy, SMOTE *(Sensitivity = 0.85, Specificity = 0.74, Accuracy = 0.74)* provided slightly better Sensitivity, whereas SMOTE + Tomek *(Sensitivity = 0.84, Specificity = 0.75, Accuracy = 0.75)* provided slightly better Specificity and Accuracy. Therefore, it can be concluded that Ensemble Voting Classifier provided better results with both SMOTE and SMOTE + Tomek when compared against Tomek Links.

Table 11 : Performance of Ensemble Voting Classifier with Different Sampling Techniques

Sr No.	Model Name	Sample Technique	Sensitivity (TPR)	Specificity (TNR)	AU ROC	Accuracy	TP	FN	FP	TN
1	Voting Classifier	SMOTE	0.85	0.74	0.79	0.74	133	24	2197	6326
2	Voting Classifier	TOMEKLINKS	0.13	0.99	0.56	0.97	21	136	108	8415
3	Voting Classifier	SMOTETOMEK	0.84	0.75	0.79	0.75	132	25	2158	6365

4.13 Comparison of the Sampling Techniques

The 70% of actual data was sampled using 3 different sampling techniques, i.e., SMOTE, Tomek Links and SMOTE + Tomek, as discussed in section 4.1.2. The models developed using the data sampled by each of these techniques were then evaluated using the remaining 30% of the data (test data).

Based on the evaluation and results in the above section, it can be observed that while evaluating the performance of Tomek Links (under-sampling) technique, on the basis of Sensitivity, Specificity and AUROC (True Negative Rate), the results obtained are biased in the favour of majority class (Negative / Healthy cases) for the models like XGBoost *(Sensitivity = 0.04, Specificity = 1, AUROC = 0.52)*, Support Vector Classifier (SVC) *(Sensitivity = 0.06, Specificity = 0.98, AUROC = 0.52)*, K-Nearest Neighbour (KNN) *(Sensitivity = 0, Specificity = 1, AUROC = 0.5)*, AdaBoost *(Sensitivity = 0, Specificity = 1,*

AUROC = 0.5) and Voting Classifier *(Sensitivity = 0.13, Specificity = 0.99, AUROC = 0.56)* as they have very high Specificity and very low Sensitivity on the contrast, indicating that all the minority cases (stroke) are predicted as majority cases (healthy). The same is reflected by very high accuracy and very high False Negative (FN) value. Also, AUROC score of near 0.5 suggests that the model is unable to distinguish between the stroke and healthy cases. Similarly, for other models like Random Forest, Neural Network, Naïve Bayes, Logistic Regression, Decision Tree the performance of Tomek Links is not better when compared against SMOTE and SMOTE + Tomek.

While comparing the performance of SMOTE and SMOTE + Tomek based on AUROC score, the performance for both the techniques is similar for almost all the models except Neural Network *(SMOTE AUROC = 0.69, SMOTE + Tomek AUROC = 0.71)* where SMOTE + Tomek performed slightly better than SMOTE. Similarly, based on accuracy both the techniques performed almost similar except for XGBoost *(SMOTE Accuracy = 0.68, SMOTE + Tomek Accuracy = 0.69)*, Neural Network *(SMOTE Accuracy = 0.73, SMOTE + Tomek Accuracy = 0.74)* and Voting Classifier *(SMOTE Accuracy = 0.74, SMOTE + Tomek Accuracy = 0.75)* where SMOTE + Tomek demonstrated slight improvement over SMOTE. Therefore, it can be concluded that SMOTE + Tomek is the best suitable sampling technique for the problem of stroke detection.

4.14 Comparison of the Developed Models

As SMOTE + Tomek is determined as the best performing sampling technique for this project post comparison in section 4.13, Models that are implemented using the data sampled by SMOTE + Tomek technique are evaluated as part of this section.

While evaluating against the AUROC performance metrics that reflect the ability of a model to distinguish between the stroke and the healthy cases, it is evident that the worst-performing model is SVC *(AUROC = 0.69)* whereas the best performing models are Logistic Regression and Voting Classifier *(AUROC = 0.79).*

Similarly, while evaluating against the Sensitivity (TPR) that reflect the correctly predicted stroke cases, Neural Network *(Sensitivity = 0.68)* turned out to be the worst performer, whereas the best performers were acclaimed by Logistic Regression and Voting Classifier with the Sensitivity of 0.85 and 0.84 respectively.

For Specificity (TNR) which indicates the correctly predicted healthy cases, XGBoost *(Specificity = 0.69)* was the worst performer. Whereas, Decision Tree and Voting Classifier *(Specificity 0.75)* were the best performers.

Likewise, for Accuracy that indicates the overall correctly predicted cases XGBoost *(Accuracy = 0.69)* again turned out to be the worst performer and Decision Tree and Voting Classifier *(Accuracy = 0.75)* turned out to be the best performer.

Therefore, it can be concluded that while evaluating all the 9 classification models and their respective Ensemble Voting Classifier against the performance metrics of Sensitivity, Specificity, AUROC and Accuracy, except Ensemble Voting Classifier, there is no single best performing model for all the performance metrics.

4.15 Conclusion

Based on the implemented data sampling techniques, developed classification models and respective results, it can be concluded that this project has successfully answered the Research Question and Sub-Research Question presented in section 1.2 and has achieved all the objectives set in section 1.3. For the problem of stroke detection, while hybrid sampling technique of SMOTE + Tomek was chosen as best performing technique with slight improvement over the over-sampling technique of SMOTE and major improvement over the under-sampling technique of Tomek Links, Ensemble Voting Classifier with SMOTE + Tomek sampling technique turned out to be the best performing model on all the evaluation metrics by correctly identifying 84% of the total stroke cases *(Sensitivity = 0.84)* and at the same time correctly identifying 75% of the total healthy cases *(Specificity = 0.75)*, therefore achieving the AUCROC score of 0.79 which indicates that the model is able to correctly distinguish 79% of the stroke and healthy cases. The outcomes of this project will contribute towards the body of knowledge and to the healthcare industry for the earlier detection of stroke.

5 Discussion

5.1 Contribution to the Stake Holders

The attained results suggest that this ICT solution was successful to address the problem of Stroke Detection. The key contribution of this solution is towards the potential Stroke Patients as well as the entire Healthcare industry. By earlier identification the patients at risk of stroke, this solution can help save lives and can also reduce the risk of permanent disabilities due to stroke. On the other hand, as this analysis is purely based on the Electronic Health Records which are accessible to all the hospitals and medical institutions, this solution has potential to open a new prospect for such institutions as they can no longer address the patients after the occurrence of a disease but can also address it proactively before the occurrence.

5.2 Challenges

Some of the key challenges that were faced as part of this project includes identifying the appropriate techniques to handle imbalanced datasets, identifying the appropriate models for ensembling along with the execution and analysis of 30 different combinations of machine learning models and sampling techniques, to evaluate the best performing model and effectiveness of Ensemble Voting Classifier model.

There were several ways identified to handle the imbalanced dataset. However, it was difficult to identify the best possible techniques. Literature was then reviewed to handle this challenge and sampling techniques were identified to be the most popular. While assessing different sampling techniques, 3 popular sampling techniques with each belonging to different type were chosen. Of which, after the evaluation, SMOTE + Tomeklinks was identified as the best way to progress with the problem of Stroke Detection. Similarly, 9 different classification models that try to cover all horizons and techniques of data analytics and machine learning were chosen for the implementation of Ensemble Voting Classifier to attain variety in the models.

Execution of 30 different combinations of machine learning models and sampling techniques was the toughest challenge to overcome however it was necessary to ensure the effectiveness of the Ensemble Voting Classifier against the individual models. The evaluation metrics of all these 30 models were added in a single dataframe and is exported in the form of .csv to ease the analysis activity.

5.3 Deliverables to the Hospitals (Stake Holders)

Post the successful implementation and evaluation of the machine learning models, it is evident that Ensemble Voting Classifier is the best performing model. Therefore, the predictions made by all the Ensemble Voting Classifiers along with their respective original features are exported to .csv files with the title "Prediction_Using_<<Sampling Technique>>.csv" using Python. The evaluation metrics values for all the 30 combinations of classification models and data sampling technique is also exported as a separate .csv file with title "Output Results.csv" to ease the analysis and maintain track of the performance over the time. Figure 9 shows the sample of .csv files exported.

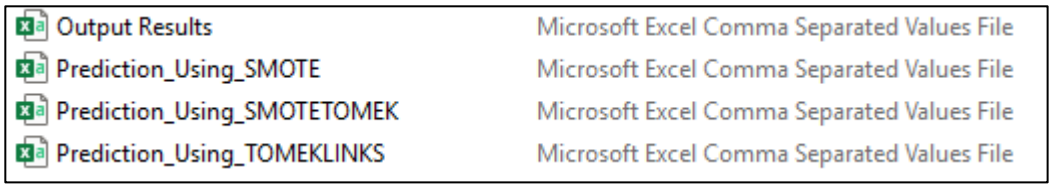

Figure 9 : CSV Deliverables

6 Conclusion and Future Work

This project aimed at developing an ICT solution that contributes towards the healthcare industry by earlier identification of stroke that is one of the major life-threatening diseases of the 21st century. To achieve the goal with the help of data analytics, 9 different machine learning models were implemented with each using 3 different data sampling techniques. These 9 models were than ensembled into an Ensemble Voting Classifier model that uses soft voting to predict the outcome based on the probability provided by those models. This Ensemble Voting Classifier model along with the SMOTE + Tomek sampling technique turned out to be the best performing model as it was successfully able to distinguish between 79% of the stroke and healthy cases *(AUCROC = 0.79)* and was successfully able to identify 84% of the overall stroke cases *(Sensitivity = 0.84)*.

Therefore, it can be concluded that by end of this technical report, this project has successfully addressed the research question presented in section 1.2, covering all the identified gaps in the related work (section 2.5) and has therefore successfully addressed the problem of stroke detection.

Future Work:

The future work of this project is to proceed with a similar approach for identifying other hazardous diseases like Cancer. It also includes identification and addition of several new features that can improve the performance further. As this project is purely based on the analysis of the Electronic Health Records and considering the odd case of the hospital not maintaining the health records electronically, future work of this project also revolves around

the implementation of an Optical Character Recognition model for conversion of such data into .csv.

Therefore, to summarise the future work, the aim is to develop a robust software suite that can be widely accessible by all the medical institutions and at the same time can help in predicting all the hazardous diseases like Stroke and Cancer.

Acknowledgement

I would like to thank my mentor Dr. Catherine Mulwa for her constant invaluable support and guidance throughout the tenure of this project.

References

Azevedo, A. & Santos, M. F., 2008. KDD, SEMMA and CRISP-DM: a parallel overview. *IADIS European Conference Data Mining,* pp. 182-185.

Batista, G. E., Prati, R. C. & Monard, M. C., 2004. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *ACM SIGKDD explorations newsletter,* 6(1), pp. 20-29.

Chang, C., Cheng, M. & Ma, M., 2018. Application of Machine Learning for Facial Stroke Detection. *In 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP),* 23(1), pp. 1-5.

Elhassan, T. & Aljurf, M., 2017. Classification of Imbalance Data using Tomek Link (T-Link) Combined with Random Under-sampling (RUS) as a Data Reduction Method. *Global J Technol Optim 2017,* 2(1), pp. 1-11.

El-Sayed, A. A., Mahmood, M. A. M., Meguid, N. A. & Hefny, H. A., 2015. Handling autism imbalanced data using synthetic minority over-sampling technique (SMOTE). *Third World Conference on Complex Systems (WCCS),* Volume 3, pp. 1-5.

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P., 1996. The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM,* 39(11).

García-Terriza, L. et al., 2019. Comparison of different machine learning approaches to model stroke subtype classification and risk prediction. *Institute of Electrical and Electronics Engineers (IEEE),* pp. 1-10.

Gulenko, A. et al., 2016. Evaluating machine learning algorithms for anomaly detection in clouds. *In 2016 IEEE International Conference on Big Data (Big Data),* pp. 2716-2721.
Guo, X. et al., 2008. On the Class Imbalance Problem. *Fourth international conference on natural computation,* 4(1), pp. 192-201.

Jeena, R. & Kumar, S., 2016. Stroke Prediction Using SVM. *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT),* pp. 600-602.

Jeon, S. et al., 2018. RISK-Sleep : Real-time Stroke Early Detection System During Sleep Using Wristbands. *IEEE International Conference on Systems, Man, and Cybernetics (SMC),* pp. 4333-4339.

McFadden, E., Luben, R. & Wareham, N., 2009. Social class, risk factors, and stroke incidence in men and women: a prospective study in the European prospective investigation into cancer in Norfolk cohort. *Stroke,* 40(4), pp. 1070-1077.

Oughali, M., Bahloul, M. & El Rahman, S., 2019. Analysis of NBA Players and Shot Prediction Using Random Forest and XGBoost Models. *2019 International Conference on Computer and Information Sciences (ICCIS),* pp. 1-5.

Popukaylo, V., 2019. Predicting the occurrence of strokes using the language R. *Computer Science Journal of Moldova,* 27(1), pp. 73-84.

Qazi, N. & Raza, K., 2012. Effect Of Feature Selection, Synthetic Minority Over-sampling (SMOTE) And Under-sampling On Class imbalance Classification. *UKSim 14th International Conference on Computer Modelling and Simulation,* 14(1), pp. 145-150.

Ren, J., 2012. ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging. *Knowledge-Based Systems,* 26(1), pp. 144-153.

Sain, H. & Purnami, S., 2015. Combine sampling support vector machine for imbalanced data classification. *Procedia Computer Science,* 72(1), pp. 59-66.

Singh, M. & Choudhary, P., 2017. Stroke Prediction using Artificial Intelligence. *Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON),* 8(1), pp. 158-161.

Sudha, A., Gayathri, P. & Jaisankar, N., 2012. Effective analysis and predictive model of stroke disease using classification methods. *International Journal of Computer Applications,* 43(14), pp. 26-31.

Troncoso, A. et al., 2018. Imbalanced classification techniques for monsoon forecasting based on a new climatic time series. *Environmental Modelling and Software,* 106(1), pp. 48-56.

Vijayalakshmi, V., Babu, M. & Lakshmi, R., 2018. KFCM Algorithm for Effective Brain Stroke Detection through SVM Classifier. *IEEE International Conference on System, Computation, Automation and Networking (ICSCA),* pp. 1-6.

Wirth, R. & Hipp, J., 2000. CRISP-DM: Towards a standard process model for data mining. *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining,* pp. 29-39.

Xie, J., Wang, Z., Yu, Z. & Guo, B., 2018. Enabling Efficient Stroke Prediction by Exploring Sleep Related Features. *IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI),* pp. 452-461.

Zeng, M. et al., 2016. *Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data.* Chongqing, IEEE International Conference of Online Analysis and Computing Science (ICOACS), pp. 225-228.