PRML Major Project on Brain Stroke Prediction

Tarun Raj Singh (B21CS076) and Aryan Himmatlal Prajapati (B21EE012)

24 April 2023

1 Introduction

This Project Problem is associated with the prediction of **stroke** in a person. A **stroke** is a medical condition caused by poor blood flow to the brain, leading to cell death and the impairment of brain function. The dataset is highly unbalanced, the positive class (Stroke) account for (approx.) **6%** of all Enteries. Also, on finding the direct correlations between the features and the classes, there is no relevant or significant correlation. In this project we have implemented a complete machine learning pipeline which at last predicts whether , given the details about a Person , he/she is likely to have a stroke or not.

2 Highly Imbalanced data :

We found that the **class 1 (Stroke)** is **19.084** times lesser than the negative **class (No Stroke).**

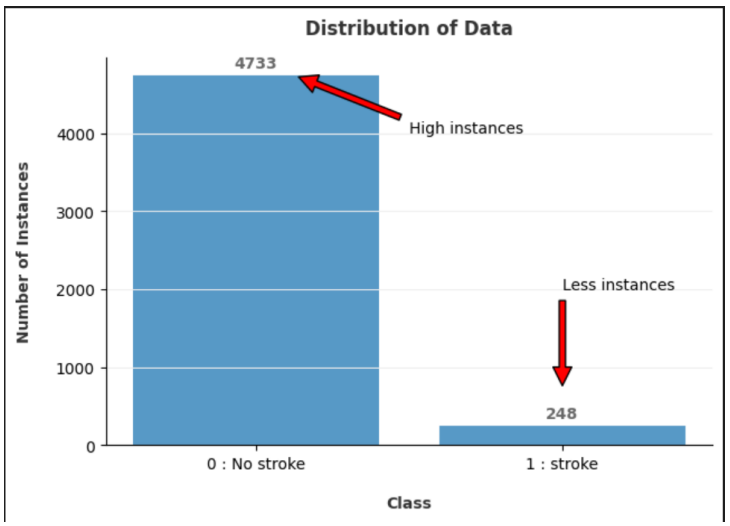


Figure 1: Bar graph showing the Highly imbalance nature of data set

Hence, any model trained on this imbalanced data is highly prone to be biased towards the **negative class** , hence we have to reduce the false negative predictions from the model using various imbalance handling methods.

3 Pre-Processing :

- Since, we found no direct significant correlations between the features and the classes, and also the dataset is not too high dimesional. Hence, using **all** the features for further tasks.
- Normalized the data , (the features with continous values) using the maximum absolute approach, to make their scale in the range 0 to 1.

- Checked the Unique values for all the object Datatype features and found that feature (smoking_status) have **1500** enteries as **Unknown** (Which means that the details about the smoking status of those people is unavailable) for inclusion in the dataset, so replacing these Unknown values with the feature **Mode** (Which is never smoked in this case). Also checked that there are no NULL values left in the data to handle.
- Finally categorically encoded all the object type features. Seperated out Features and Labels.

4 If applying Ensemble methods on this imbalanced data :

We Tried to use **AdaBoost** , **XGBoost** and **LGBM** Models on this Imbalanced data and found that the Models are actually biased towards the negative class. This is clear from the fact that they are performing well for the negative class , but the overall f1 score for the positive class is very low .

- **For Adaboost : f1 = 0.03** , (*Misclassify 61 (out of 62) Stroke Instances*)
- **For XGBoost : f1 = 0.05** , (*Misclassify 60 (out of 62) Stroke Instances*)
- **For LGBM : f1 = 0.05** , (*Misclassify 60 (out of 62) Stroke Instances*)

But in all these model we are getting an **accuracy close to 94%** . This is due to the biased nature of the classifier towards the negative class and instances of negative class are far far more in number, hence the accuracy is more , but it is not a good model as it can not separate out Stroke instances correctly . So we need to apply some **imbalance handling methods** and then train the classifiers .

i.e , Although the model is working well for classifying the class 0 **(No Stroke)** instance , still the Precision , Recall and F1 score for class 1 is not good enough , this is due to the highly imbalanced data which is biased towards predicting class 0 , To improve this , we can apply various Imbalance handling algorithms.

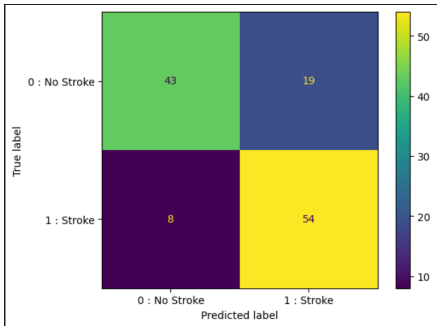
5 Applying Imbalance handling methods using Adaboost :

5.1 Using UnderSampling :

UnderSampling is a technique in Data Analysis where the size of the majority class is reduced to balance the class distribution. Hence, we randomly sampled **248** instances from the negative class and added them with the **248** instances of Positive class to create a balanced dataset and then applied the **AdaBoost classifier.**

	precision	recall	f1-score	support
0 : No Stroke	0.84	0.69	0.76	62
1 : Stroke	0.74	0.87	0.80	62
accuracy			0.78	124
macro avg	0.79	0.78	0.78	124
weighted avg	0.79	0.78	0.78	124

(a) Classification report



(b) Confusion Matrix

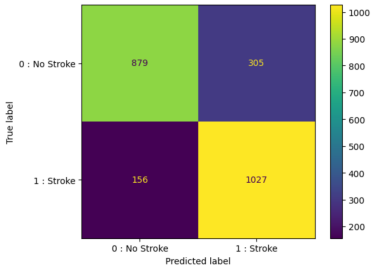
Figure 2: Results of AdaBoost Using Under Sampling

5.2 Using OverSampling :

OverSampling is a technique in Data Analysis where the size of the minority class is increased to balance the class distribution. Hence, we randomly sampled **4733** instances from the positive class with replacement and added them with the **4733** instances of negative class to create a balanced dataset and then applied the **AdaBoost classifier.**

	precision	recall	f1-score	support
0 : No Stroke	0.85	0.74	0.79	1184
1 : Stroke	0.77	0.87	0.82	1183
accuracy			0.81	2367
macro avg	0.81	0.81	0.80	2367
weighted avg	0.81	0.81	0.80	2367

(a) Classification report



(b) Confusion Matrix

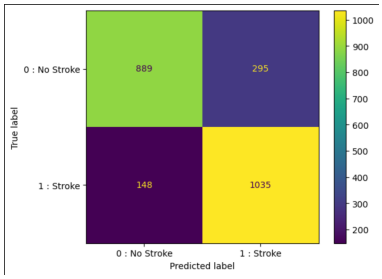
Figure 3: Results of AdaBoost Using Over Sampling

5.3 Using SMOTE (Synthetic Minority Over Sampling Technique) :

SMOTE (Synthetic Minority Over Sampling Technique) is a method for generating synthetic samples to balance class distribution in imbalanced datasets . Hence, we synthetically sampled **4733** instances from the positive class and added them with the **4733** instances of negative class to create a balanced dataset and then applied the **AdaBoost classifier**.

	precision	recall	f1-score	support
0 : No Stroke	0.86	0.75	0.80	1184
1 : Stroke	0.78	0.87	0.82	1183
accuracy			0.81	2367
macro avg	0.82	0.81	0.81	2367
weighted avg	0.82	0.81	0.81	2367

(a) Classification report



(b) Confusion Matrix

Figure 4: Results of AdaBoost Using SMOTE Sampling

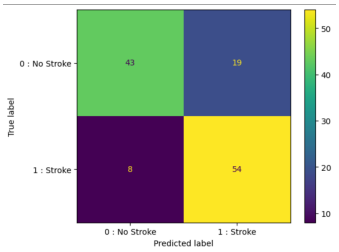
6 Applying Imbalance handling methods using XGBoost :

6.1 Using UnderSampling :

Applied the **XGBoost** Classifier on the previouly generated **undersampled** balanced dataset.

	precision	recall	f1-score	support
0 : No Stroke	0.84	0.69	0.76	62
1 : Stroke	0.74	0.87	0.80	62
accuracy			0.78	124
macro avg	0.79	0.78	0.78	124
weighted avg	0.79	0.78	0.78	124

(a) Classification report



(b) Confusion Matrix

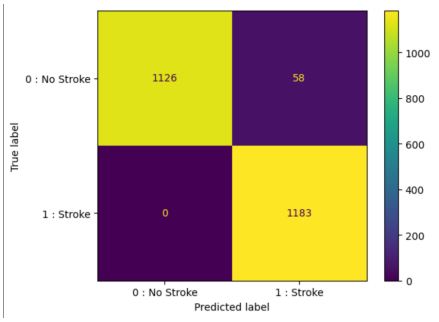
Figure 5: Results of XGBoost Using Under Sampling

6.2 Using OverSampling :

Applied the **XGBoost** Classifier on the previouly generated **oversampled** balanced dataset.

	precision	recall	f1-score	support
0 : No Stroke	1.00	0.95	0.97	1184
1 : Stroke	0.95	1.00	0.98	1183
accuracy			0.98	2367
macro avg	0.98	0.98	0.98	2367
weighted avg	0.98	0.98	0.98	2367

(a) Classification report



(b) Confusion Matrix

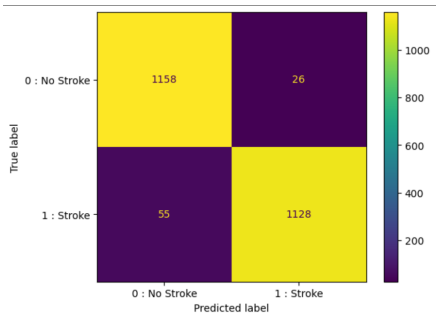
Figure 6: Results of XGBoost Using Over Sampling

6.3 Using SMOTE (Synthetic Minority Over Sampling Technique) :

Applied the **XGBoost** Classifier on the previouly generated Synthetically sampled balanced dataset using **SMOTE** .

	precision	recall	f1-score	support
0 : No Stroke	0.95	0.98	0.97	1184
1 : Stroke	0.98	0.95	0.97	1183
accuracy			0.97	2367
macro avg	0.97	0.97	0.97	2367
weighted avg	0.97	0.97	0.97	2367

(a) Classification report



(b) Confusion Matrix

Figure 7: Results of XGBoost Using SMOTE Sampling

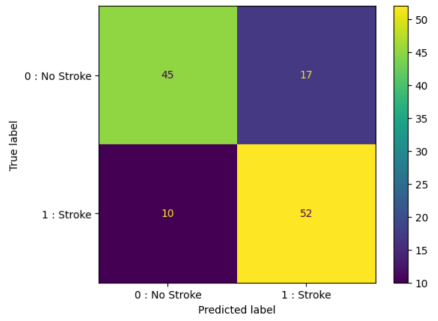
7 Applying Imbalance handling methods using LGBM :

7.1 Using UnderSampling :

Applied the **LGBM** Classifier on the previouly generated **undersampled** balanced dataset.

	precision	recall	f1-score	support
0 : No Stroke	0.82	0.73	0.77	62
1 : Stroke	0.75	0.84	0.79	62
accuracy			0.78	124
macro avg	0.79	0.78	0.78	124
weighted avg	0.79	0.78	0.78	124

(a) Classification report



(b) Confusion Matrix

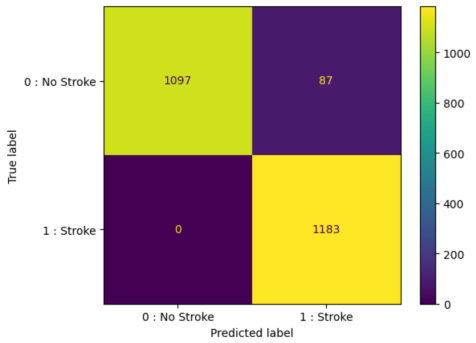
Figure 8: Results of LGBM Using Under Sampling

7.2 Using OverSampling :

Applied the **LGBM** Classifier on the previouly generated **oversampled** balanced dataset.

	precision	recall	f1-score	support
0 : No Stroke	1.00	0.93	0.96	1184
1 : Stroke	0.93	1.00	0.96	1183
accuracy			0.96	2367
macro avg	0.97	0.96	0.96	2367
weighted avg	0.97	0.96	0.96	2367

(a) Classification report



(b) Confusion Matrix

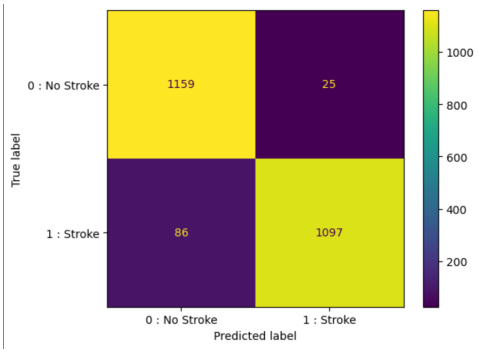
Figure 9: Results of LGBM Using Over Sampling

7.3 Using SMOTE (Synthetic Minority Over Sampling Technique) :

Applied the **LGBM** Classifier on the previouly generated Synthetically sampled balanced dataset using **SMOTE** .

	precision	recall	f1-score	support
0 : No Stroke	0.93	0.98	0.95	1184
1 : Stroke	0.98	0.93	0.95	1183
accuracy			0.95	2367
macro avg	0.95	0.95	0.95	2367
weighted avg	0.95	0.95	0.95	2367

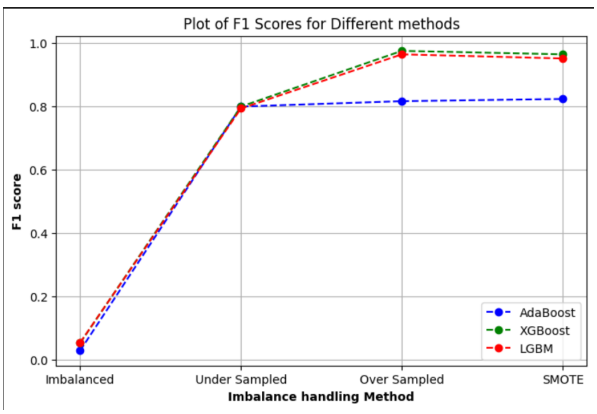
(a) Classification report



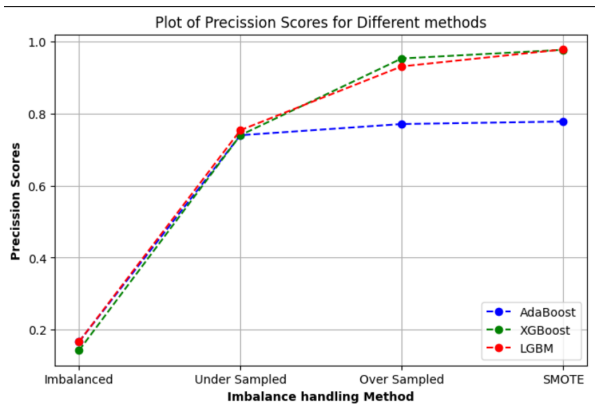
(b) Confusion Matrix

Figure 10: Results of LGBM Using SMOTE Sampling

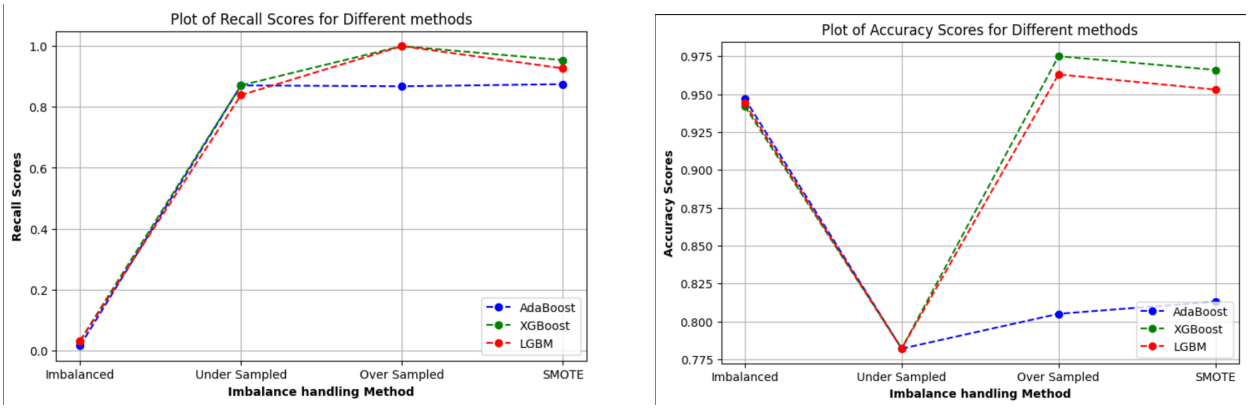
8 Plots for Performance on different model selections and Different Balancing Methods :



(a) F1 Score of Different Methods



(b) Precision Score of Different Methods



(a) Recall Scores of Different Methods

(b) Accuracy Scores of Different Methods

Figure 11: Performances of all models and different Balancing methods.

9 Picking out the best model till Now :

We used all these **boosting algorithms** in order to reduce bias and used sampling methods to reduce the imbalance in the dataset and finally tried different configurations and observed the confusion matrix. Different Models were performing differently , so we have taken that model which is having most of the **metrics** as **high**.

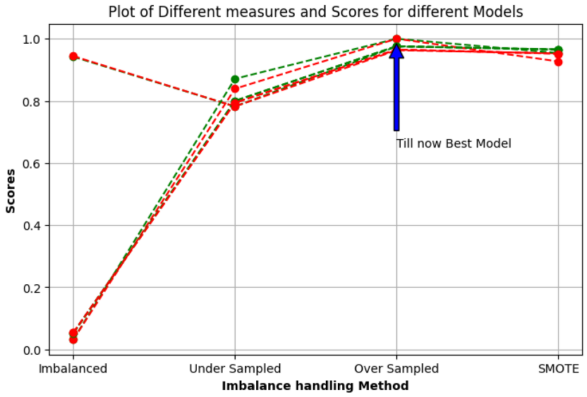


Figure 12: Plot for all metrics and all models in order to observe the best model

Chose the best model till now as **Oversampling model using XGBoost**.

10 Tarun and Aryan Sampling Method :

10.1 Motivation :

Till Now , we have tried various methods to handle the imbalance in the data and train a model which is well suited to predict the **Minority** Positive class (Class 1 : **Stroke**) .

Till Now , the model which performed best is the **XGBoost** model with the randomly oversampled dataset. Still , it is misclassifying some instance from the Majority Class (Class 0 : No stroke) , this could be due to the fact that the nature of the prediction problem and the dataset is in a way **biased** towards the Class 0 , but we have oversampled the minority class to make a ratio of 1:1 for both the classes. This somehow, giving more importance to the positive minority class but on the same hand **supressing** the majority nature of the majority class.

Hence , we will try to **oversample** the **minority class** plus the **majority class** as well, to put some more importance on the majority class as well.

10.2 Sampling Method :

Since, the original ratio of the Number of instance in the minority positive class (Class 1 : Stroke) and the Majority negative class (Class 0 : No stroke) is in the ratio **1 : 20** (approx).

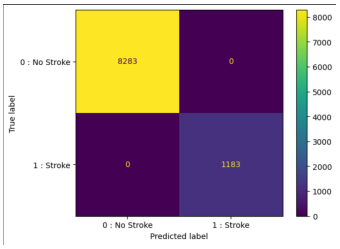
And ,in the above work we saw that the original ratio **1 : 20** is performing very bad by suppressing the minority class and the Balanced Oversampling method using the ratio **(1 : 1)**, is supressing the majority nature of the Class 0 . Hence, we tried the ratios between these ratios and checked that the **ratio (1:7)**, performs best for the dataset. Hence training the model, with **1 : 7 ratio Oversampled data** (which means oversample **class 1 , 1 times** and **class 0 , 6 times** (**with replacement**)).

11 Results with Tarun and Aryan Sampling :

As mentioned above , we used our own sampling method and then applied the **XGBoost** classifier on it. The Classification report and the confusion matrix for the test set of the dataset generated using our sampling for this Highly Imbalanced data and using the XGBoost Classifier:

	precision	recall	f1-score	support
0 : No Stroke	1.00	1.00	1.00	8283
1 : Stroke	1.00	1.00	1.00	1183
accuracy			1.00	9466
macro avg	1.00	1.00	1.00	9466
weighted avg	1.00	1.00	1.00	9466

(a) Classification report



(b) Confusion Matrix

Figure 13: Results of XGBoost Classifier Using Tarun and Aryan Sampling on test data

Thus, we observed that the model is well trained for both the classes and Performed very well for the test data.

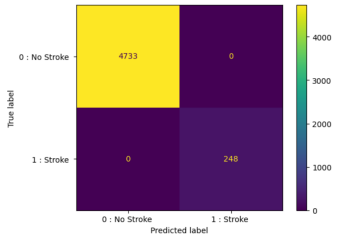
No Stroke Instances Predicted as Stroke (*Out of 8,283 test instances*) : **0**
Stroke Instances Predicted as No Stroke (*Out of 1,183 test instances*) : **0**
Total Misclassified Instances (*Out of 9,466 test instances*) : **0**

12 Let's Check Results on Complete original highly imbalanced dataset :

We used out best trained model using our own sampling to predict the complete **actual highly imbalanced dataset**.
The Classification report and the confusion matrix :

	precision	recall	f1-score	support
0 : No Stroke	1.00	1.00	1.00	4733
1 : Stroke	1.00	1.00	1.00	248
accuracy			1.00	4981
macro avg	1.00	1.00	1.00	4981
weighted avg	1.00	1.00	1.00	4981

(a) Classification report



(b) Confusion Matrix

Figure 14: Results of Our best trained model on complete original highly imbalanced data

Thus, we can see that our best trained model is correctly classifying all the points in the original **Highly Imbalance dataset**.

13 Conclusion :

Our best trained model is correctly classifying all the **Stroke** and **Non stroke** instances of the actual data , hence the model is not **biased** towards any of the classes , this can be seen from the fact that it performed the same on the test data for the **Oversampled XGBoost**, It did not classified any Stroke instances as Non stroke. So, every Stroke instance is caught with an accuracy of **almost 100%** and every Non Stroke instance is detected corectly with an accuracy of **almost 100%**

Hence using the above mentioned **Balancing techniques** and **Classifiers** we handled the bias and imbalance in the data set and got a classifier trained which can handle both classes **efficiently** and without missing Stroke instances .

14 Contributions :

We both sat together for the whole project work at each and every step. Hence, the whole project is a combined effort. We, did not divide the project amongst us in parts. Although , we sat together for generating the .ipynb file and the report together, **Tarun Raj Singh (B21CS076)** wrote the python code for the ipynb file and **Aryan Himmatlal Prajapati (B21EE012)** wrote the Latex code for the Report. All other sampling methods and training classifiers are a combined effort of both of us.

Thank You