Stroke Detection: Prediction with Multiple Classification Models

Group 13 Project Final Report

University of Wisconsin - Madison
Computer Science 539 - Introduction to Artificial Neural Networks

Tony Wang - ywang3298@wisc.edu
College of Letters and Science - Undergraduate
Hanchi Chen - Hchen794@wisc.edu
College of Letters and Science - Undergraduate

Submission date: Dec.14, 2023

Abstract

The goal of this project is to use multiple machine learning classification algorithms to predict whether a patient has a stroke based on several parameters like gender, age, various diseases, and smoking status. We have rigorously explored our dataset, implementing preprocessing measures and conducting a thorough visualization and analysis. Following baseline research, we successfully validate an example algorithm on Kaggle[1]. Subsequently, we select and implement several models including Support Vector Classification (SVC), Logistic Regression, Random Forest, and XGBoost. These models were further refined using k-fold cross-validation for optimal performance. The performance of these algorithms was evaluated using the confusion matrix, AUC, recall, and other relevant metrics.

1 Introduction

1.1 Problem Statement

According to the data from the World Health Organization (WHO) in 2019, stroke is the second leading cause of death, which is responsible for approximately 11% of total deaths respectively[2]. What's more, every 40 seconds, someone in the United States has a stroke. Every 3 minutes and 14 seconds, someone dies of stroke[3]. Stroke should be paid attention to more than thought.

The current workflow when patients want to check if they have a stroke typically begins with a series of examinations, including Computerized tomography (CT), Blood tests, physical exams, and so on. Not only is this approach time-consuming, but it also poses significant financial burdens. Rather than focusing predominantly on stroke treatment, the emphasis should shift towards prevention, especially considering the abrupt onset of stroke symptoms which often renders patients incapacitated, and unable to seek timely assistance.

Machine learning algorithms can analyze large amounts of data and identify patterns and correlations that can be used to make predictions with a high degree of accuracy. We aim to optimize the predictive power of various machine-learning classifiers for stroke detection. The ultimate goal will be finding an effective model to predict if the patient has a stroke or not.

1.2 Performance Metrics Selection

For disease detection problems, the samples will always be imbalanced and disproportional, and negative cases may be ten to hundred times more than positive cases, which is the exact situation in the real world. In this task, stroke detection, we will have an imbalanced dataset which will be introduced in detail later. Therefore, we consider AUC a better metric to evaluate the performance in minority cases. Also, for diseases including stroke, we consider false negative is much more serious than false positive cases, based on this principle, we may sacrifice precision for a better f1-score and recall.

1.3 Related Works

In fact, many recent studies focus on training deep learning models for medical purposes, for instance, brain tumor detection and heart attack detection, are already successfully used as efficient and friendly methods. A promising solution for stroke detection may also lie in leveraging neural networks to analyze pertinent data, thereby predicting individuals at heightened risk of experiencing a stroke. For stroke detection, some existing work already reached more than 90 percent predictive accuracy in Kaggle. For example, RachidYZ achieved more than 95 percent accuracy by the Decision Tree Classifier in Figure 1.[1] This result is considered as a baseline research for us but we found that the other metrics including recall and f1 were not satisfactory, as we illustrated before, the model

may not be a valid algorithm for stroke detection. We do more tasks to improve other important and meaningful metrics other than accuracy.

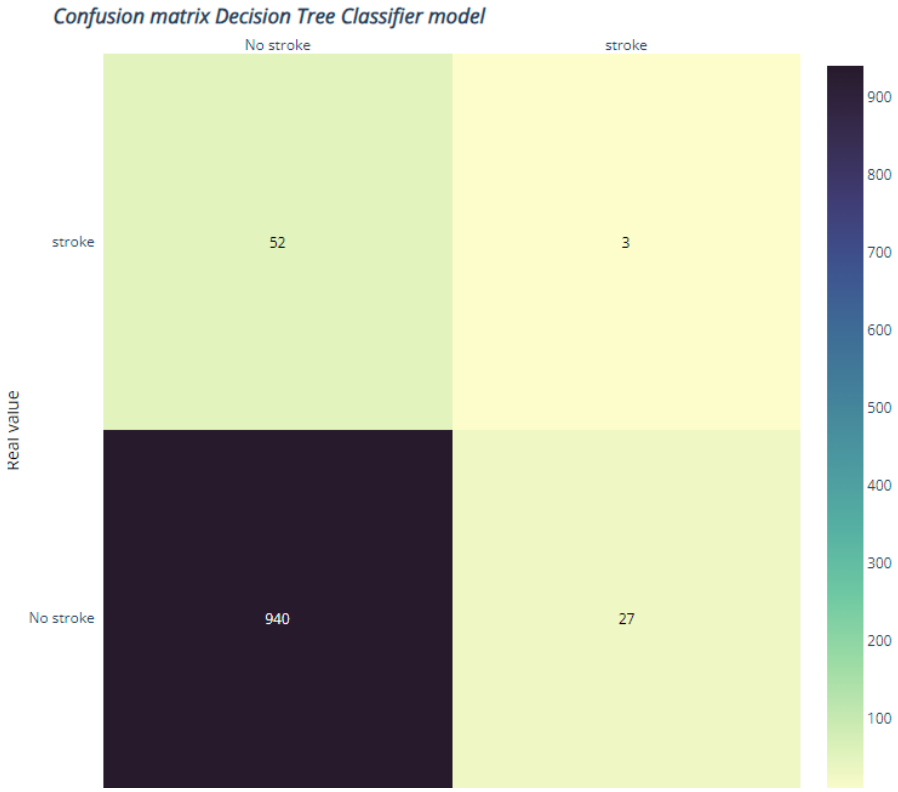


Figure 1.

2 Data

2.1 Datasets

We found the data from https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data. The dataset contains 5110 observations. There are 12 features in the original dataset and some missing data for BMI. We conduct data analysis to check the correlation of each feature to the target, and currently decided just to drop out the "id" feature since it does not influence detection. We conducted an 80/20 split of the dataset and test performance metrics against the 20% test sets. For data pre-processing, we use many techniques like Label-Encoder for encoding, SMOTE for data imbalance, imputation for missing data, and feature scaling for normalized data.

S.No.	Attribute	Code given	Value ranges	Data type
1	Gender	gender	1, 0	Binary
2	Age	age	in years	Numeric
3	Hypertension binary feature	hypertension	1, 0	Binary
4	Heart disease binary feature	heart_disease	1, 0	Binary
5	Has the patient ever been married	ever_married	1, 0	Binary
6	Work type of the patient	work_type	0, 1, 2	Nominal
7	Residence type of the patient	Residence_type	0, 1	Binary
8	Average glucose level in blood	avg_glucose_level	55.1–272	Numeric
9	Body Mass Index	bmi	10.3-97.6	Numeric
10	Smoking status of the patient	smoking_status	0,1,2	Nominal
11	Stroke event	stroke	0,1	Binary

Table 1: Description of Dataset Attributes

2.2 Data Analysis and Pre-processing

The following image shows basic information of features of this dataset, with 5110 valid samples.

```
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   gender             5110 non-null   object
 1   age                5110 non-null   float64
 2   hypertension       5110 non-null   int64
 3   heart_disease      5110 non-null   int64
 4   ever_married       5110 non-null   object
 5   work_type          5110 non-null   object
 6   Residence_type     5110 non-null   object
 7   avg_glucose_level  5110 non-null   float64
 8   bmi                4909 non-null   float64
 9   smoking_status     5110 non-null   object
 10  stroke             5110 non-null   int64
dtypes: float64(3), int64(3), object(5)
memory usage: 439.3+ KB
```

Figure 2, Dataset Basic Information.

1. Data Imputation: We use KNNImputer to deal with 201 missing data in BMI, and for the "Unknown" state in the feature "Smoking status of the patient", we assume it is a virtual state and encode it separate from other states.

2. Data Imbalance: This dataset includes ten times more negative samples than positive ones, which is common in real-world diseases. However, This can be a problem because the algorithm might only learn from the majority of examples and not pay enough attention to the minority examples, thus the model can not perform well to predict. Therefore, we use SMOTE to fix this problem by creating new, fake examples of the minority type so that they become more evenly represented in the dataset. After Oversampling, we get a new dataset with 7788 samples, which includes 3894 stroke samples and 3894 healthy samples.

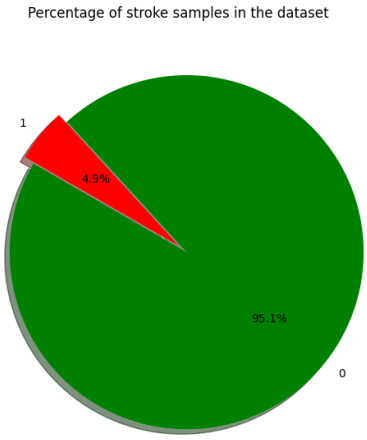


Figure 3, Imbalance Sample.

3. Data Importance: We test the correlation of each feature to stroke in the following image. The main factor to stroke is 'Age', which is significant and intuitive here. However, other factors are not influential enough and are similarly important, which may cause problems in prediction.

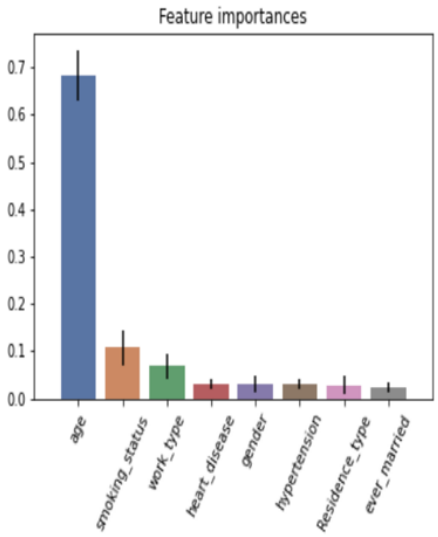


Figure 4, Data Importance.

The following analysis for detailed features referenced the code in baseline research[1].

For 'work type' and 'smoke status', they seem to be less significant for stroke, since the stroke ratios among all jobs and smoking groups are surprisingly similar.

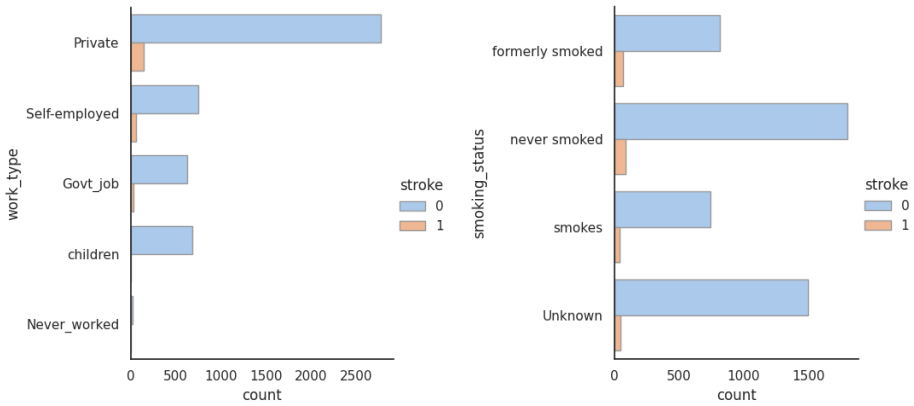


Figure 5.

We separate the stroke samples of different genders when comparing the stroke density of some features, which shows that gender is less discriminate in stroke detection. Patients in Urban places are more likely to have stroke. Meanwhile, hypertension and heart disease increase the possibility of stroke.

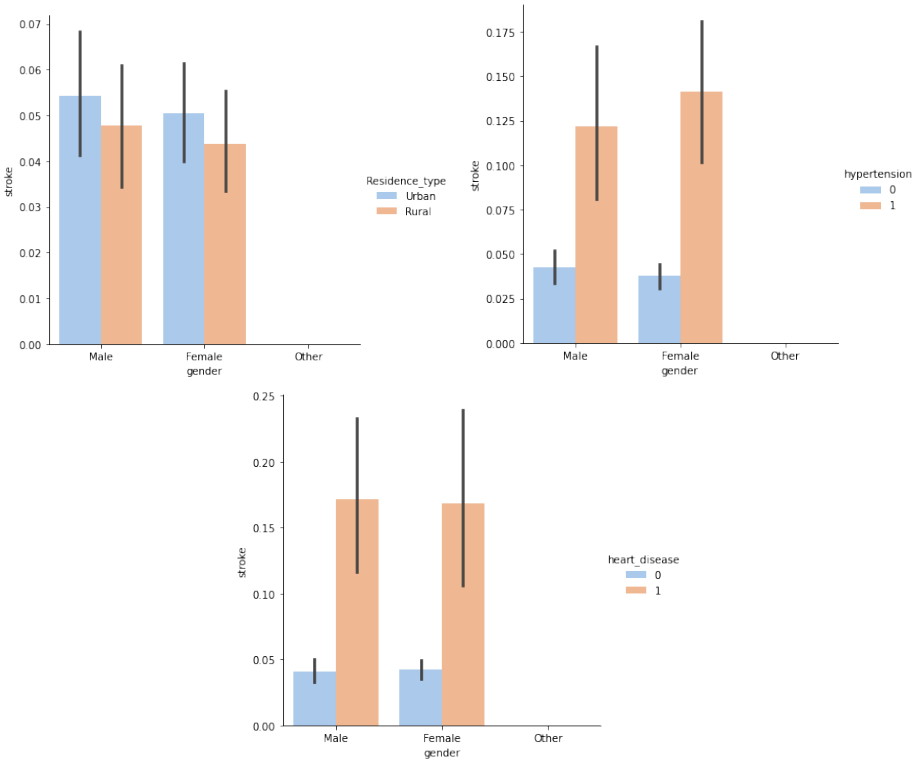


Figure 6.

4. Data encoding: We use LabelEncoder to encode the categorical columns: like 'gender', 'work type', and 'smoking status'.

5. Data scaling: We noticed that these data are not standard, for instance, after encoding, features like heart disease become binary integers, but bmi can be 100 times larger than binary. This may cause some problems with model training. However, we used the StanderdScalar in sklearn to standardize the features, but we didn't get good performance, so we gave up scaling currently.

3 Tasks Performed

3.1 Method

Stroke prediction models predict an individual's risk of having a stroke based on various factors such as age, blood pressure, cholesterol levels, smoking status, and other medical history. Machine learning algorithms can analyze large amounts of data and identify patterns and correlations that can be used to make predictions with a high degree of accuracy. Our aim is to optimize the predictive power of various machine-learning classifiers for stroke detection.

For baseline research, RachidYZ in Kaggle achieved an accuracy of more than 95 percent.[1] We considered it as a baseline example result but this model is still insufficient because the performance mostly relies on predicting negative blindly. Performance metrics other than accuracy are very low. For this project, we test multiple models such as Logistic regression, Support Vector Classifier (SVC), Random Forest, and XGBoost. We give up implementing specific Artificial Neural Network (ANN) models because the performance of existing machine learning models is already useful enough. We compare the AUC, recall, and other metrics for stroke detection. Also, we made a mistake at the beginning when we used SMOTE before the data partition, which caused data leakage, and this is also the mistake made by the baseline research.

We also utilize k-fold cross-validation for models for further performance improvement. Cross-validation is a more robust technique that involves partitioning the dataset into k-folds, and currently, we choose 5-fold considering computation and performance. Cross-validation provides a more reliable estimate of the model's performance and helps to reduce the risk of over-fitting.

Also, we fine-tuned the hyper-parameters including the max depth of Random Forest, the learning rate of XGBoost, and the K for cross-validation.

3.2 Platform

In our project, we primarily utilize GitHub for version control and Google Colab for collaborative development of our IPython Notebooks (.ipynb files). Colab's cloud-based environment facilitates simultaneous access and simplifies the code execution environment, eliminating the need for local setup.

The integration of GitHub ensures a clear version history, enhancing the maintainability and scalability of our project. Additionally, we are considering incorporating Google Cloud Platform (GCP) for its advanced computational resources and flexible data storage and processing capabilities.

4 Results and Discussions

Our primary objective is to design an efficient model for stroke detection by exploring a wide spectrum of classification models and determining the most proficient model. To evaluate our model's performance, we will use confusion matrices and AUC as our performance metrics. Also, past works in this field have set a high benchmark in terms of prediction accuracy. Given these implementations, our project aspires to achieve, or even surpass, a predictive accuracy of at least 90 percent with valid AUC and recall.

We also use additional performance matrices like f1-score, precision, ROC, etc. ROC (Receiver Operating Characteristic) and AUC (Area Under the ROC Curve) are evaluation metrics commonly used in machine learning for binary classification problems. We use ROC/AUC as evaluation metrics for binary classification models because they provide a more comprehensive view of the model's performance than accuracy alone in the imbalanced dataset.

4.1 Performance

The following two tables show the performance of four models with and without cross-validation. Therefore, we could compare the results and explore the effect of cross-validation, which increases our performance largely. We also provide images to show the main performance metrics, the confusion matrix, and the ROC/AUC for each model without cross-validation.

Based on Table 3, XG Boost achieves the highest accuracy at 87 percent, which nears the

performance of baseline research, but the real metrics like recall and AUC surpass the baseline research significantly. We also notice that the performance of SVC is limited to around 70-80 percent. The future direction to improve performance may be to scale the data properly and do more tasks to deal with data imbalances.

Model	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Random Forest				0.73
0 (Class)	0.98	0.72	0.83	
1 (Class)	0.14	0.80	0.24	
Logistic Regression				0.75
0 (Class)	0.97	0.76	0.85	
1 (Class)	0.13	0.64	0.22	
XGBoost				0.79
0 (Class)	0.97	0.81	0.88	
1 (Class)	0.14	0.56	0.23	
SVC				0.73
0 (Class)	0.97	0.74	0.84	
1 (Class)	0.11	0.58	0.19	

Table 2: Model Performance Comparison without Cross-Validation

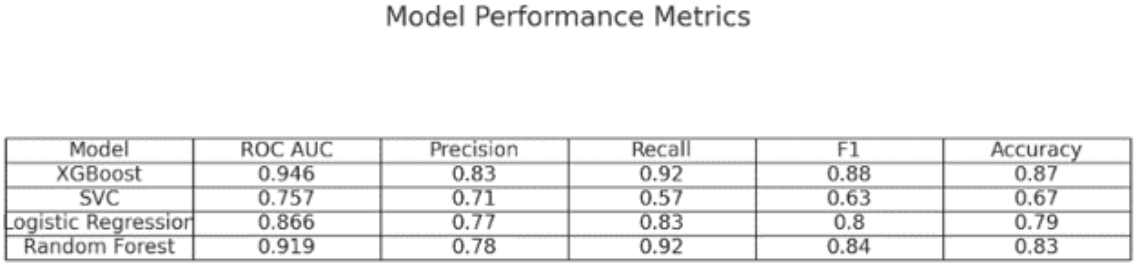
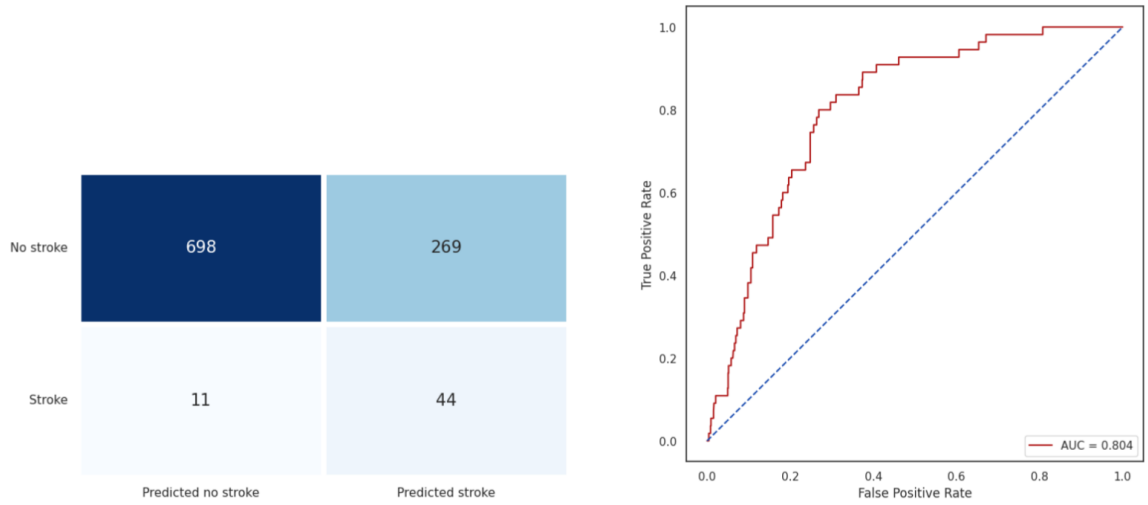


Table 3. Model Performance Comparison with 5-fold Cross-Validation

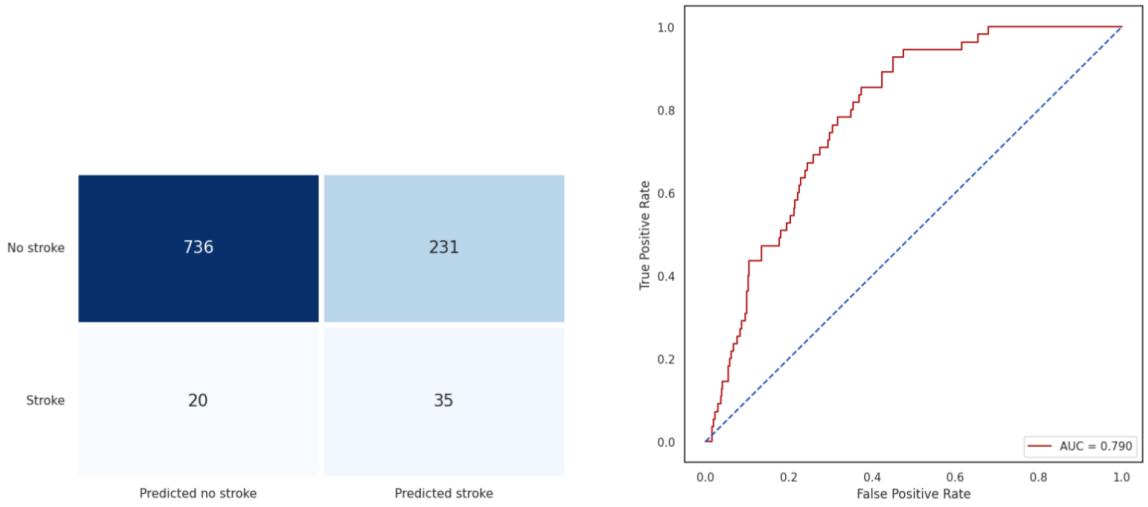
1. Logistic Regression



(a) Confusion Matrix (b) ROC/AUC

Figure 7. Results of Logistic Regression.

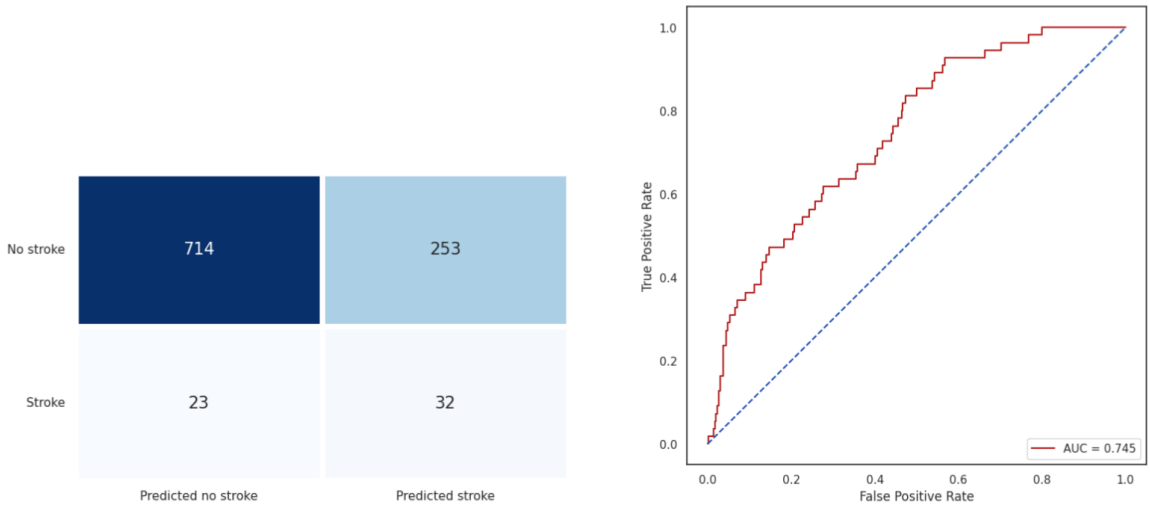
2. Random Forest



(a) Confusion Matrix (b) ROC/AUC

Figure 8. Results of Random Forest.

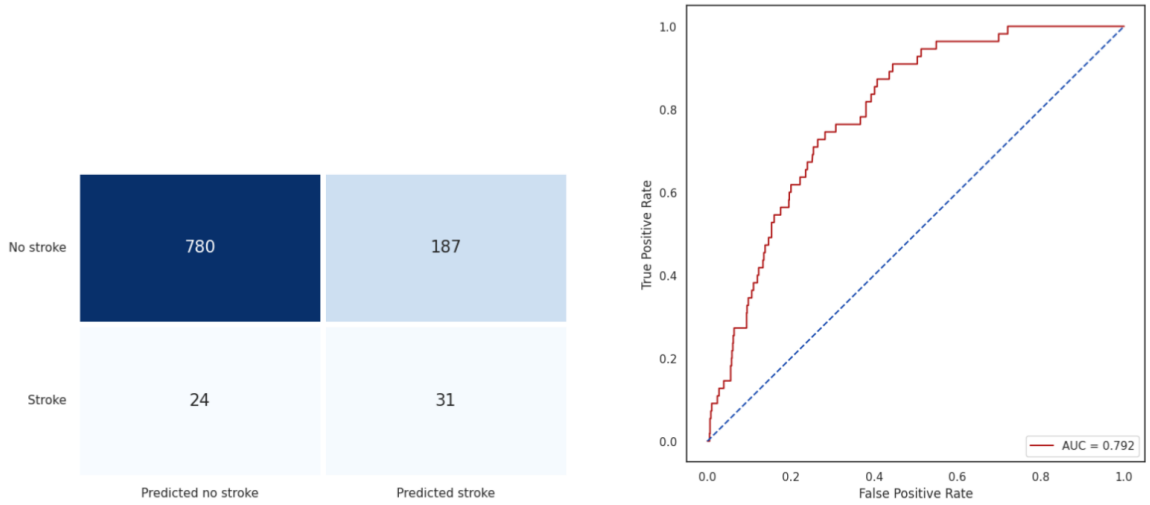
3. Support Vector Machine



(a) Confusion Matrix (b) ROC/AUC

Figure 9. Results of SVM.

4. XG Boost



(a) Confusion Matrix (b) ROC/AUC

Figure 10. Results of XG Boost.

4.2 Discussions

Four models perform well with cross-validation. For example, XGBoost shows a high AUC of 94.6% and high recall (92%). But without cross validation, there's a significant imbalance in class performance, with poor precision (14%) and f1 (23%) for the minority class. This could be due to data imbalance, bad configuration with model parameters, or insufficient features. The overall accuracy is high, but the true positive rate still can be improved, and precision may also limit the f1-score. These results highlight challenges in handling imbalanced datasets.

Critical factors are attributed to the inherent class imbalance within our dataset and the imbalance value between features. The minority class, being underrepresented, has led to the model's inadequate learning of its characteristics, thus impairing its predictive capability on the validation set. Also, the model may pay over attention to those features that have large original values.

Furthermore, the application of SMOTE on our test samples has notably influenced the outcomes. SMOTE is allowed to be employed for training datasets only, and test data must be original data in case of data leakage. In this context, SMOTE may have inadvertently introduced a bias towards the dataset, thereby skewing the model's ability to generalize on unseen data. This underscores the need for careful consideration when applying over-sampling techniques, as they can significantly affect the model's performance, especially in the context of highly imbalanced datasets.

In light of these findings, future work should focus on optimizing the model parameters specifically tailored to address data imbalance. Also, more advanced algorithms and better hyperparameters will also be explored in the future.

5 References

[1]RACHIDYZ. (2020, May 22). EDA and modeling for predicting stroke. Kaggle. Retrieved October 19, 2023, from https://www.kaggle.com/code/rachidyz/eda-and-modeling-for-predicting-stroke.

[2]"The Top 10 Causes of Death." World Health Organization, World Health Organization, www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. Accessed 21 Oct. 2023.

[3]"How Many People Are Affected by/at Risk for Stroke?" Eunice Kennedy Shriver National Institute of Child Health and Human Development, U.S. Department of Health and Human Services, www.nichd.nih.gov/health/topics/stroke/conditioninfo/risk. Accessed 21 Oct. 2023.