Stroke Prediction

by Guldanika Osmonova

Machine Learning

Table of Contents

Introduction

Stroke is a significant global health concern, ranking as one of the leading causes of death and long-term disability. The impact of stroke is profound, affecting millions of individuals and imposing a substantial burden on healthcare systems worldwide. Early detection and prevention are crucial, as timely intervention can mitigate the severe consequences associated with stroke, such as physical and cognitive impairments. This project aims to address this critical need by developing a predictive model that assesses the likelihood of stroke in individuals based on relevant patient data.

The primary objective of this project is to leverage machine learning techniques to create a reliable tool for early stroke prediction. By analyzing patient data, including age, gender, medical conditions, and lifestyle factors, the model seeks to identify individuals at high risk of experiencing a stroke. This predictive capability enables healthcare professionals to implement preventive measures, such as lifestyle modifications, medication, or enhanced monitoring, thereby reducing the incidence of stroke and improving patient outcomes.

The relevance of this project to healthcare is underscored by its potential to transform stroke prevention strategies. In resource-limited settings, predictive models offer a cost-effective means to prioritize care and allocate medical resources efficiently. By shifting the focus from treatment to prevention, this project aspires to contribute to reducing the global burden of stroke, ultimately saving lives and enhancing the quality of life for at-risk individuals.

In summary, this project not only highlights the importance of stroke prediction in modern healthcare but also demonstrates the potential of data-driven approaches to improve health outcomes. The subsequent sections will delve into the data, methodology, and results, providing a comprehensive overview of the project's execution and findings.

Data Description

The dataset utilized in this stroke prediction project was sourced from Kaggle, specifically from the "Stroke Prediction Dataset" by fedesoriano. It comprises 5,110 entries, each representing an individual patient's profile with 12 attributes. These attributes are crucial for developing a predictive model to assess stroke risk.

The dataset includes the following key features:

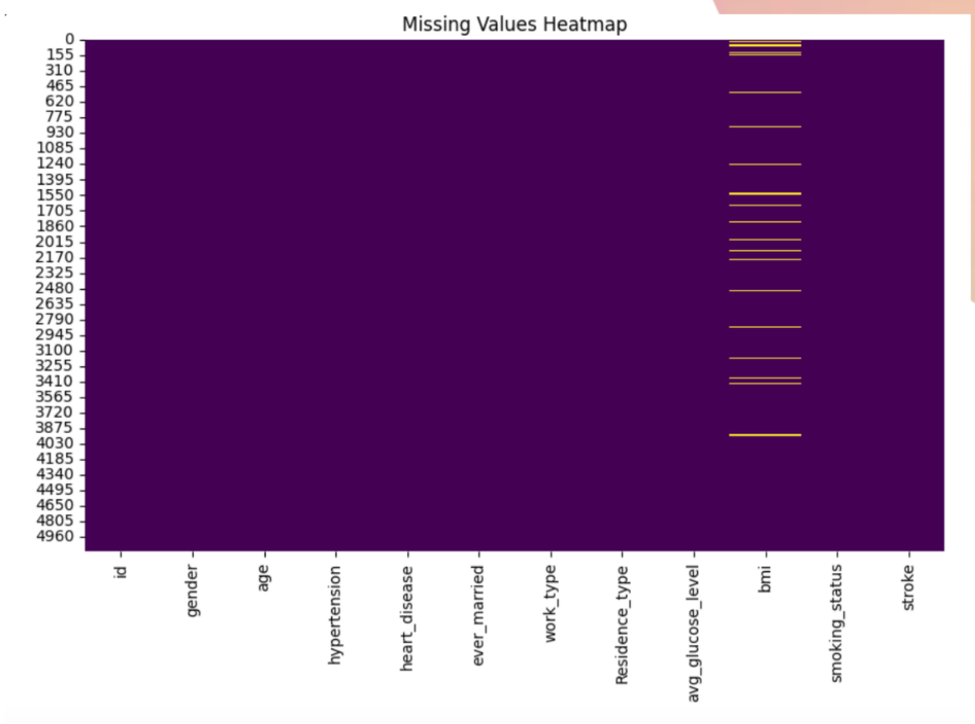
- id: A unique identifier for each patient.
- gender: Categorical variable indicating the patient's gender, with possible values "Male," "Female," or "Other."
- age: A continuous variable representing the patient's age.
- hypertension: A binary variable where 0 indicates the absence of hypertension and 1 indicates its presence.
- heart_disease: A binary variable where 0 indicates no heart disease and 1 indicates the presence of heart disease.
- ever_married: Categorical variable indicating marital status, with values "No" or "Yes."
- work_type: Categorical variable describing the patient's type of employment, with categories including "children," "Govt_job," "Never_worked," "Private," and "Self-employed."
- Residence_type: Categorical variable indicating whether the patient resides in a "Rural" or "Urban" area.
- avg_glucose_level: A continuous variable representing the average glucose level in the blood.
- bmi: A continuous variable representing the body mass index of the patient.
- smoking_status: Categorical variable with values "formerly smoked," "never smoked," "smokes," or "Unknown," where "Unknown" indicates unavailable information.
- stroke: The target variable, a binary indicator where 1 signifies that the patient has experienced a stroke and 0 indicates no stroke.

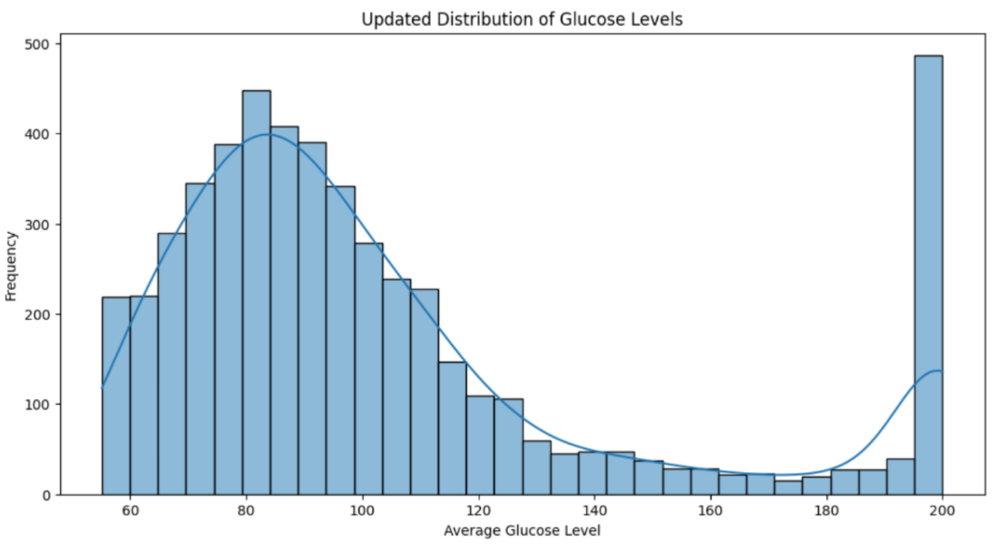
The dataset was collected with the aim of providing comprehensive patient profiles to facilitate the development of a predictive model. However, it is important to note that the dataset has some limitations, such as missing values in the 'bmi' and 'smoking_status' attributes, which were addressed through imputation techniques. Additionally, the dataset exhibits class imbalance, with a significantly higher number of non-stroke cases compared to stroke cases, which necessitates careful consideration during model evaluation and development. These attributes, along with their respective data types and distributions, were thoroughly explored during the exploratory data analysis phase to ensure the integrity and suitability of the data for predictive modeling. This foundational understanding of the dataset is critical as we proceed to the methodology section, where data preprocessing and model development are discussed in detail.

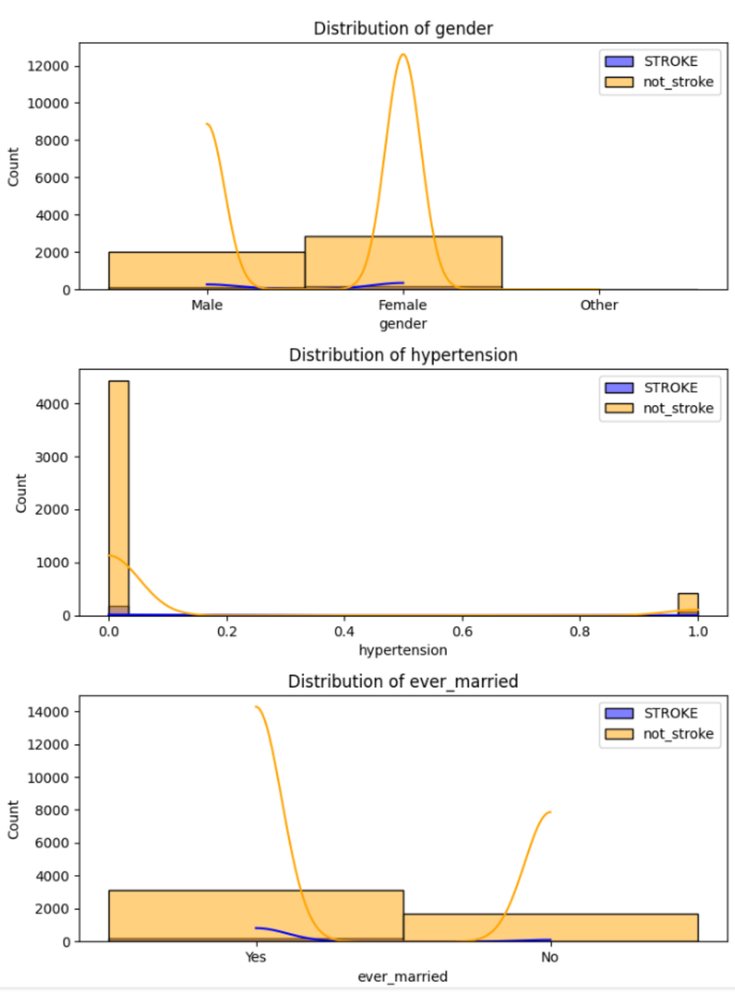
Methodology

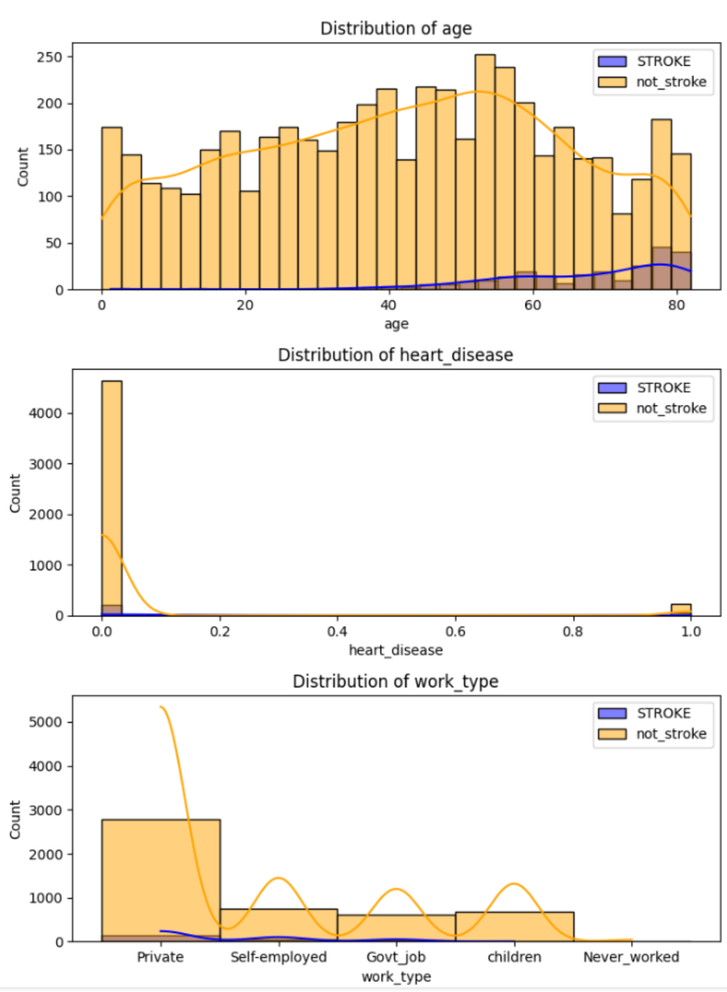
Systematic Approach

The methodology section outlines the systematic approach employed in developing the stroke prediction model, detailing the processes from data preprocessing to model development. Initially, data preprocessing involved handling missing values, data cleaning, and transformation techniques to ensure the dataset's integrity and suitability for analysis. Feature selection and engineering were conducted to enhance the model's predictive capabilities by identifying and constructing relevant attributes. Subsequently, the model development phase focused on selecting appropriate machine learning algorithms, training and validating the models, and employing evaluation metrics to assess their performance. This structured methodology aim to build a robust predictive model capable of accurately assessing stroke risk, thereby contributing to effective healthcare interventions.









The model was trained on a dataset with imbalanced classes, with a majority of instances belonging to the 'no stroke' class. To address this, the SMOTE technique was employed to balance the classes during the training phase.

Data Preprocessing

The data preprocessing phase is a critical step in preparing the dataset for effective model development. This process involves several key tasks aimed at ensuring the data is clean, consistent, and suitable for analysis.

Initially, I addressed missing values within the dataset. The 'bmi' attribute was identified as having missing entries, which were handled using median imputation. This approach was chosen due to its robustness against outliers and its ability to preserve the central tendency of the data without introducing bias. The median is particularly effective for continuous variables, making it a suitable choice for the 'bmi' column.

```
# handling missing values
df['bmi'] = df['bmi'].fillna(round (df['bmi'].median(), 2))
df.isnull().sum()
```

	0
id	0
gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
bmi	0
smoking_status	0
stroke	0

dtype: int64

```
Outlier bounds for 'bmi': lower=10.30, upper=46.30
Outliers in 'bmi':
          id  gender   age  hypertension  heart_disease ever_married  \
21     13861  Female  52.0             1              0          Yes
66     17004  Female  70.0             0              0          Yes
113    41069  Female  45.0             0              0          Yes
254    32257  Female  47.0             0              0          Yes
258    28674  Female  74.0             1              0          Yes
...      ...     ...   ...           ...            ...          ...
4906   72696  Female  53.0             0              0          Yes
4952   16245    Male  51.0             1              0          Yes
5009   40732  Female  50.0             0              0          Yes
5057   38349  Female  49.0             0              0          Yes
5103   22127  Female  18.0             0              0           No

          work_type Residence_type  avg_glucose_level   bmi   smoking_status  \
21    Self-employed          Urban             233.29  48.9     never smoked
66          Private          Urban             221.58  47.5     never smoked
113         Private          Rural             224.10  56.6     never smoked
254         Private          Urban             210.95  50.1          Unknown
258   Self-employed          Urban             205.84  54.6     never smoked
...             ...            ...                ...   ...              ...
4906        Private          Urban              70.51  54.1     never smoked
4952  Self-employed          Rural             211.83  56.6     never smoked
5009  Self-employed          Rural             126.85  49.5  formerly smoked
5057      Govt_job          Urban              69.92  47.6     never smoked
5103        Private          Urban              82.85  46.9          Unknown
```

Following the imputation, I conducted an exploratory analysis to detect potential outliers, particularly in the 'bmi' and 'avg_glucose_level' attributes. The Interquartile Range (IQR) method was employed to identify extreme values that could skew the analysis. This step ensures that the data fed into the model is representative of typical patient profiles, thereby enhancing the model's predictive accuracy.

Next, I transformed categorical variables to a numerical format suitable for machine learning algorithms. One-hot encoding was applied to categorical variables with more than two unique values, such as 'work_type' and 'smoking_status'. This technique prevents the introduction of ordinal relationships where none exist, thus maintaining the integrity of the categorical data. For binary categorical variables like 'gender' and 'ever_married', label encoding was utilized, converting these attributes into numerical form while preserving their binary nature.

MODELLING

1. Splitting Data into Training and Testing Sets

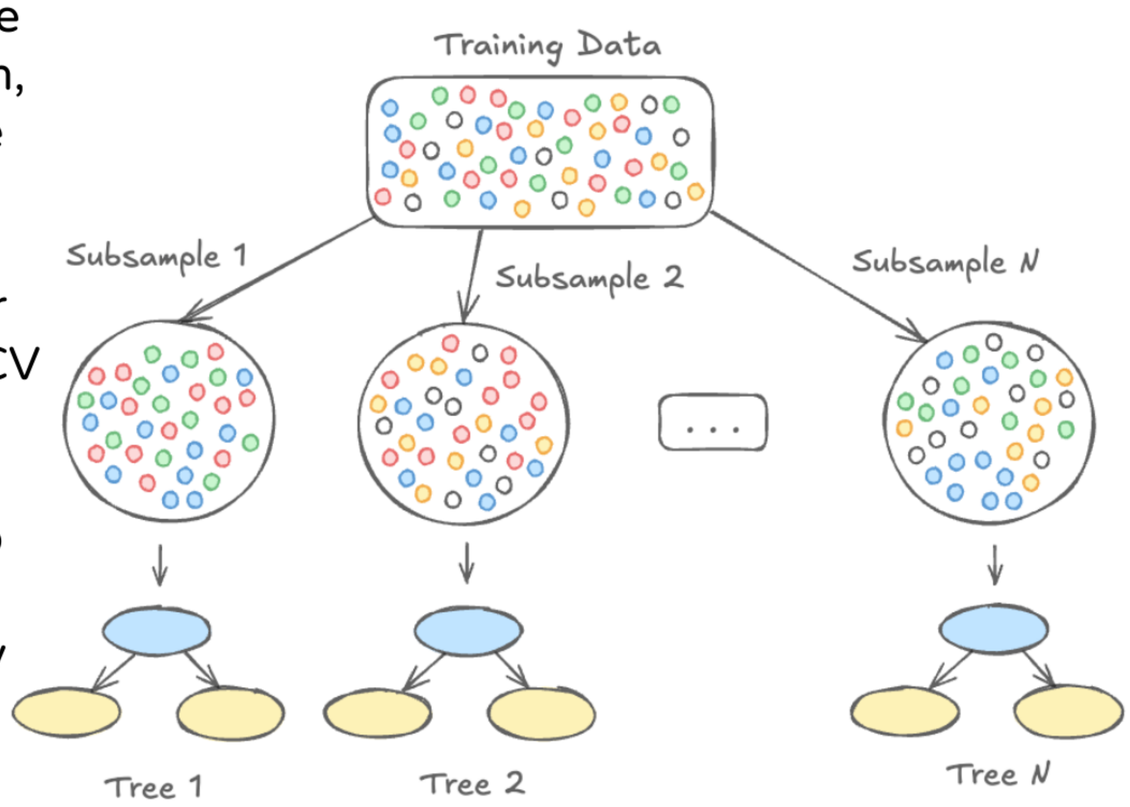
```
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE

# Split data into features and target
X = df.drop(columns=['stroke'])
y = df['stroke']

# Train-Test Split (80% train, 20% test), stratified to keep the target class distribution
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# Apply SMOTE to balance the training dataset
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
```

Additionally, I addressed the issue of class imbalance in the dataset, which is a common challenge in medical datasets where the occurrence of the event of interest (stroke, in this case) is relatively rare. The Synthetic Minority Over-sampling Technique (SMOTE) was employed to balance the classes by generating synthetic samples for the minority class. This step is crucial to prevent the model from being biased towards the majority class and to improve its ability to detect strokes accurately.

Model building

In the development of the predictive model for stroke risk assessment, a comprehensive approach was undertaken to select and evaluate various machine learning algorithms. The primary objective was to identify a model that could effectively predict stroke occurrences, given the significant class imbalance in the dataset. The dataset, sourced from Kaggle, was split into training and testing sets to facilitate model training and evaluation. The target variable, 'stroke', was imbalanced, with a majority of instances belonging to the 'no stroke' class. To address this, Synthetic Minority Over-sampling Technique (SMOTE) was employed to balance the classes during the training phase.

Several machine learning algorithms were considered, including Logistic Regression, Random Forest, Support Vector Machine (SVM), Gradient Boosting, XGBoost, and CatBoost. Each model was trained using the training dataset, and hyperparameter tuning was performed using GridSearchCV to optimize model performance. The evaluation of these models was primarily based on recall, given the critical need to minimize false negatives in a healthcare setting. Recall was chosen as the primary metric to ensure that the model could identify as many true stroke cases as possible, thereby reducing the risk of undetected strokes.



	Accuracy	ROC-AUC	Recall
Logistic Regression	0.726917	0.741697	0.758065
Random Forest	0.934272	0.521567	0.064516
Gradient Boosting	0.928013	0.579504	0.193548
XGBoost	0.939750	0.524446	0.064516
SVM	0.703443	0.583948	0.451613

Since stroke prediction is an imbalanced classification problem, RECALL and ROC-AUC is more informative than accuracy. A high accuracy may simply reflect the model's bias toward the majority class.

The XGBoost model emerged as the most effective in terms of recall, achieving a recall score of 0.8871, which was significantly higher than other models. Despite its lower accuracy, XGBoost demonstrated a strong ability to detect stroke cases, making it the preferred choice for this application. Additionally, the ROC-AUC metric was used as a secondary evaluation criterion to assess the model's overall discriminative ability. The XGBoost model achieved a ROC-AUC score of 0.7886, indicating a moderate level of performance in distinguishing between stroke and non-stroke cases.

The final model was saved using joblib for future deployment and evaluation. However, the low recall on the test set suggests that further improvements are necessary. Future efforts may involve enhancing data quality, exploring additional features, or employing advanced techniques to further improve model sensitivity and robustness. This leads into the discussion of the results and their implications for healthcare practice.

EVALUATION

The performance of the predictive model was evaluated using multiple machine learning algorithms. The evaluation metrics focused on accuracy, recall, precision, F1-score, and ROC-AUC, with particular emphasis on recall and ROC-AUC due to the imbalanced nature of the dataset.

The XGBoost model achieved the highest accuracy at 93.97%, but its recall for the stroke class was low, indicating a bias towards the majority class.

The CatBoost model showed a more balanced performance with a recall of 33.87% and a ROC-AUC of 0.7796.

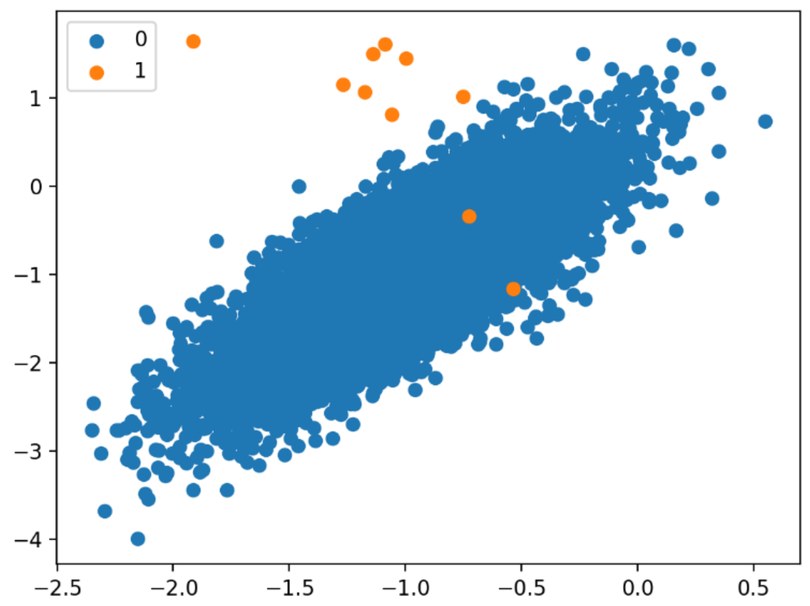
The SVM model achieved a moderate recall of 45.16%, but with a lower overall accuracy of 70.34%, highlighting the trade-off between sensitivity and specificity.

CHALLENGES

This stroke prediction project highlights the challenges of developing a reliable healthcare model for early stroke detection. The primary goal was to identify individuals at high risk of stroke to enable timely interventions. However, performance metrics, particularly recall scores, reveal that the model struggles to accurately identify stroke cases, with class imbalance being a major issue. Despite using techniques like SMOTE to address this, the model shows bias towards the majority class, limiting its effectiveness in detecting the minority class of stroke cases, which is crucial for early intervention.

The limitations of the model are further compounded by the choice of features and algorithms. While relevant attributes like age, hypertension, and heart disease were included, they may not provide enough predictive power for accurate stroke identification. Additionally, linear models such as logistic regression, along with ensemble methods like Random Forest and XGBoost, did not perform significantly better, suggesting the need for more advanced techniques or additional data. To improve the model's sensitivity, future work should focus on acquiring higher-quality data, incorporating more relevant features, and exploring advanced machine learning techniques, such as deep learning.

RESULTS



Class Imbalance

The model struggled to identify stroke cases due to the class imbalance.

Model Limitations

The model may lack strong predictive power due to the limited features.

Clinical Applicability

The model is not yet suitable for deployment in a clinical setting.

References

- World Health Organization. (2023). Stroke: A leading cause of death and disability. Retrieved from https://www.who.int/news-room/fact-sheets/detail/stroke
- Fedesoriano. (2023). Stroke Prediction Dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.
- Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Advances in Neural Information Processing Systems (pp. 3146-3154).
- Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
- Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20(3), 273-297.
- Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 29(5), 1189-1232.
- Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences, 55(1), 119-139.
- McKinney, W. (2010). Data Structures for Statistical Computing in Python. In Proceedings of the 9th Python in Science Conference (pp. 51-56).
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.
- Waskom, M. L. (2021). Seaborn: Statistical Data Visualization. Journal of Open Source Software, 6(60), 3021.