PREDICTING STROKE RISK: A MULTI-ALGORITHM APPROACH USING MACHINE LEARNING

Satya Kiran Kota **Akshit Sharma**
Sai Srinivas Munagala **Nitish Kumar Pinneti**

Abstract

This mid-term report encapsulates our ongoing project aimed at revolutionizing the prediction of cerebral strokes using machine learning techniques. Utilizing a comprehensive dataset from Kaggle, we conducted exploratory data analysis to understand patterns, imbalances, and associations within the data. We've made significant strides, including data preprocessing and implementing logistic regression model. Our future focus lies in enhancing recall by exploring different classification algorithms, ensemble learning, and feature engineering.

1. Introduction

India is grappling with a significant health crisis, with stroke being a leading cause of mortality and disability. The urgency of early detection and intervention is underscored by the fact that strokes are the fourth leading cause of death in the country(Organization, 2020). Disruptions in blood flow to the brain can have severe consequences, making it critical to promptly identify warning signs. However, early detection and treatment are key to minimizing damage and improving outcomes.

Our efforts have centered on gathering and preparing data for analysis. We extracted information from various sources and converted it into a format suitable for analysis. Using Exploratory Data Analysis, we visually explored the data, identifying and addressing any anomalies or missing information. Through this process, we ensured the quality and reliability of our dataset, which is crucial for building accurate prediction models.

With the prepared data, we implemented logistic regression and decision tree machine learning models from scratch to understand their performance. These models serve as a foundation for our ongoing work where we plan to explore more advanced models like SVM, decision tree(Quinlan, 1986), voting classfiers(Wang et al., 2013) along with techniques oversampling(Mohammed et al., 2020), feature selection. By refining our models, we aim to provide actionable insights for proactive stroke prediction in India, contributing to improved health outcomes.

2. Methodology

This section includes a dataset description, and details regarding cleaning and pre-processing of the dataset.

2.1. Data Collection

The dataset utilized for this project comprises cerebral stroke prediction details sourced from *Kaggle*(Federico, 2021). It consists of *12* columns, with *43,400* entries, no duplicate records were identified. However, null values were detected in the *bmi* and *smoking_status* columns, with *1,462* and *13,292* occurrences respectively.

Table 1. Description of Dataset being utilized.

COLUMN NAME	DATA TYPE	NULL COUNT
ID	$int64$	0
GENDER	$object$	0
AGE	$float64$	0
HYPERTENSION	$int64$	0
HEART_DISEASE	$int64$	0
EVER_MARRIED	$object$	0
WORK_TYPE	$object$	0
RESIDENCE_TYPE	$object$	0
AVG_GLUCOSE_LEVEL	$float64$	0
BMI	$float64$	1462
SMOKING_STATUS	$object$	13292
STROKE	$int64$	0

2.2. Data Cleaning

2.2.1. HANDLING NULL VALUES

Given the substantial presence of null values (*25% of rows*), removing them could compromise dataset integrity. To address this, exploratory data analysis, including plotting column occurrences, was conducted to understand dependencies and stroke correlations. For the *bmi* column, normal imputation via median replacement was employed due to outliers detected via box plot analysis. Conversely, for *smoking_status*, given the categorical nature of entries, null values

were imputed using the *unknown* category, introducing a new data dimension.

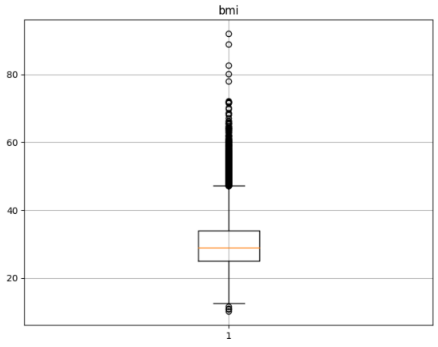

Figure 1. Boxplot analysis of *bmi* column of Dataset, showing the outliers.

2.2.2. Cleaning the Dataset

After mitigating null values, redundant columns were identified and removed. The *id* column, lacking any evident impact on stroke prediction, was removed. All remaining columns were retained.

2.3. Exploratory Data Analysis

During exploratory data analysis (EDA), we examined the distribution of each feature with respect to the stroke value, analyzing the occurrence of each value within each column. Additionally, we investigated the dependencies between two features by utilizing correlation and covariance matrices. This allowed us to gain insights into potential patterns, class imbalances, anomalies, and associations within the data.

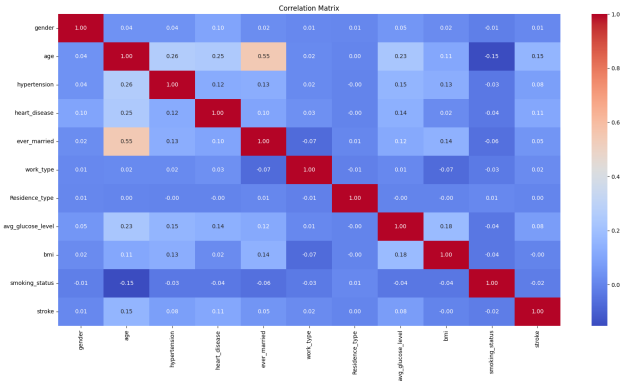

Figure 2. EDA: The correlation matrix of the columns.

2.4. Preprocessing

2.4.1. Label Encoding

Label encoding is done to convert categorical variables into numerical format, which is required by many machine learning algorithms. It assigns a unique integer to each category, allowing algorithms to interpret the data numerically. However, label encoding introduces ordinality, implying an order or rank, which may not always be appropriate for categorical variables with more than two categories.

2.4.2. Test and Train Split

The dataset was partitioned into distinct sections for training and testing to ensure model evaluation and performance assessment. Data shuffling was performed to evenly distribute samples, with *25%* of the data reserved for testing purposes. The remaining dataset was allocated for training ML models. The random state for shuffling was set to *42*.

2.4.3. Sampling Techniques

Following dataset splitting, it was observed that the training dataset exhibited bias, potentially leading to skewed model outcomes. To mitigate this, various sampling techniques were explored, focusing on addressing class imbalance.

Considering the significant disparity in class distribution within the training dataset (*31975* instances of *non-stroke* (0) and *575* instances of *stroke* (1)), under sampling methods could result in data loss due to the vast difference in sample sizes.

So, over sampling methods were deemed more suitable. Two prominent techniques, Random Over Sampling and Synthetic Minority Over-sampling Technique (*SMOTE*), were considered. We plan to implement this oversampling in further progress on the project.

3. Progress

3.1. Work Done So Far

The project has made significant progress since its inception. Data collection involved obtaining a comprehensive dataset from *Kaggle*, containing *43,400* entries with *12* columns. Initial preprocessing steps were undertaken to address data quality issues, including the identification and handling of null values in the *bmi* and *smoking_status* columns. Exploratory data analysis (EDA) was conducted to gain insights into the dataset's characteristics and dependencies. Challenges such as class imbalance and outlier detection were addressed during this phase.

Algorithm 1 Logistic Regression

Initialize: $\theta_j = 0$ for all $0 \leq j \leq m$, size m
repeat
$\underline{\hat{y}} = sigmoid(X \cdot \underline{\theta} + \underline{b})$
$d\theta = \frac{1}{n}[X^T \cdot (w_1 \underline{y}(1 - \underline{\hat{y}}) + w_0(1 - \underline{y})\underline{\hat{y}})]$
$db = \sum_{i=1}^{n}(w_1 \underline{y}(1 - \underline{\hat{y}}) + w_0(1 - \underline{y})\underline{\hat{y}})$
$\theta = \theta - \eta d\theta$
$b = b - \eta db$
until k *iterations*

3.2. Logistic Regression Implementation

Logistic regression was employed as one of the initial algorithms to predict the likelihood of stroke based on various demographic, lifestyle, and medical factors.

3.2.1. Input and Output Representation

Model Inputs: The dataset consisted of n rows and m columns, denoted as X, where X is an $n \times m$ matrix representing the input features.

Actual Output: Denoted as $\underline{y}$, it was an $n \times 1$ vector containing the true labels for each data point.

Predicted Output: Denoted as $\underline{\hat{y}}$, it was also an $n \times 1$ vector representing the model's predictions.

Weights and Bias: The logistic regression model utilized weights ($\underline{\theta}$) of size $m \times 1$ and a bias term (b).

Learning rate: The logistic regression uses η as learning rate, it's a hyper parameter.

3.2.2. Loss Function and Optimization

Initially, the model underwent training using gradient descent optimization to minimize the *standard binary cross-entropy loss* function.

Upon observation of Fig. 2, it became apparent that the accuracy reached 98%. However, in the context of an imbalanced dataset, accuracy alone does not adequately capture model performance. Notably, the confusion matrix depicted in Fig. 2 illustrates that all instances are predicted as 0s, highlighting the limitations of accuracy as a metric.

Given the dataset's class imbalance, we subsequently opted for a weighted loss function to better address this issue.

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

The *weighted loss function* was introduced to address this imbalance, where the number of instances of the negative class (0s) was significantly higher than the positive class (1s).

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[w_0(y_i \log(\hat{y}_i)) + w_1((1-y_i)\log(1-\hat{y}_i))]$$

Formula for class weights: $w_j = \frac{n_{samples}}{(n_{classes} \times n_{samples_j})}$

3.2.3. Learning Rate

A lower learning rate leads to slower training times, while a higher learning rate can cause oscillations in the loss value. Therefore, it is crucial to find an optimal learning rate that balances faster training rates with better performance. After experimenting with various learning rates, we determined that a learning rate of 1 is optimal. A learning rate of 10 resulted in oscillations, while a rate of 0.001 required more iterations to train the model.

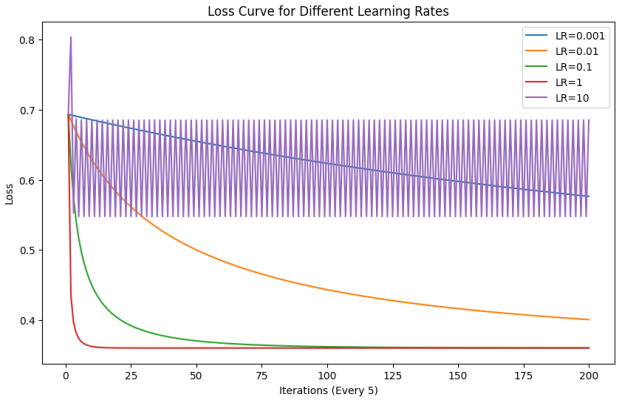

Figure 3. Loss curve at different learning rates.

3.2.4. Regularization

After conducting experiments with L1, L2, and no regularization, we observed that there was no significant change in the performance of the model during testing. Consequently, we decided not to employ any form of regularization.

3.2.5. Threshold Adjustment for Improved Performance

Given the importance of recall in stroke detection, manual thresholding was performed to optimize recall.

Manual Thresholding: This involved adjusting the threshold value for class prediction to achieve a higher recall without compromising overall accuracy.

Threshold Optimization: Through manual thresholding, a threshold of 0.0061 was identified to yield a balance between recall and accuracy. Iterative adjustment and validation on the test set were performed to determine the optimal threshold.

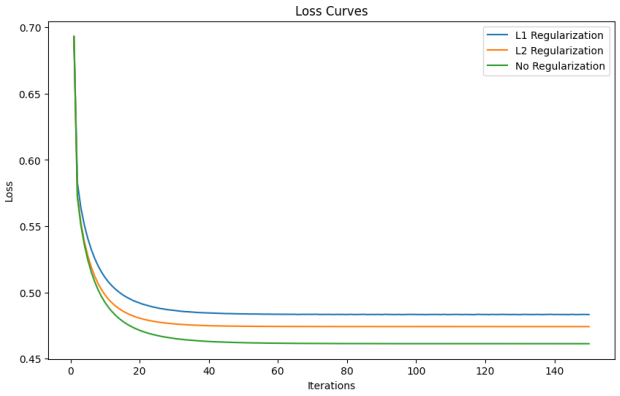

Figure 4. Regularization vs Loss Curve.

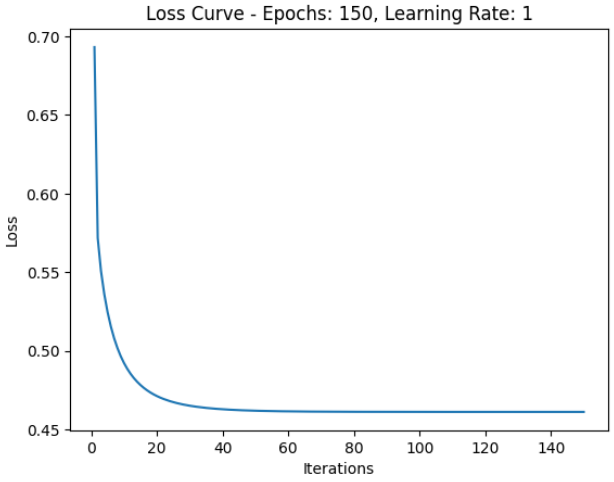

Figure 5. MSE vs iterations.

After applying manual thresholding and employing the weighted loss function, the final confusion matrix, presented in Fig. 7, was obtained. Despite exhibiting lower accuracy compared to the model using the actual loss function, this approach proves superior in stroke prediction. Notably, it successfully classifies some instances of 1s, enhancing its utility for this task. Additionally, the comparison with the logistic regression model from the scikit-learn library revealed similar outcomes, thus validating the efficacy of our logistic regression implementation.

3.3. Future Directions

To further improve recall, alternative methods such as implementing different classification algorithms or utilizing advanced techniques like ensemble learning and feature engineering will be explored.

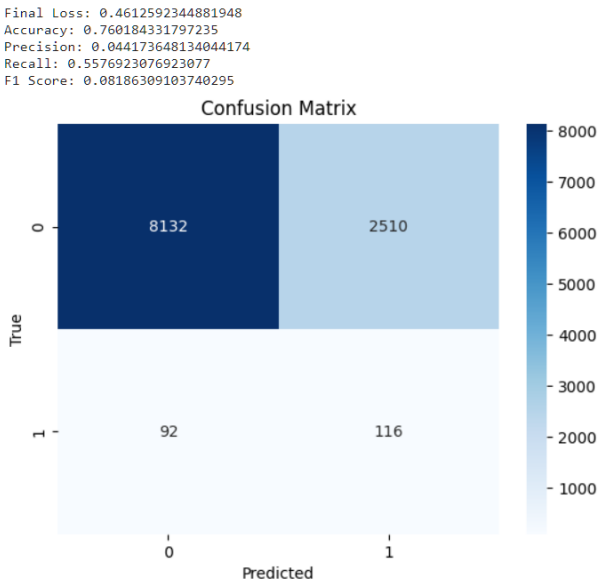

Figure 6. Confusion Matrix for weighted loss function with optimal threshold.

4. Conclusion

Thus far, we have successfully completed foundational tasks, including exploratory data analysis (EDA), data cleaning, preprocessing, and the independent implementation of Logistic Regression from scratch. Moving forward, our focus will be on implementing oversampling, SVMs, decision trees, and voting classifiers from scratch to evaluate their effectiveness in stroke prediction. With diligent implementation and thorough evaluation, our objective is to enhance prediction accuracy and make a valuable contribution to stroke risk assessment.

References

Dritsas, E. and Trigka, M. Stroke risk prediction with machine learning techniques. *Sensors (Basel)*, 22(13):4670, 2022. doi: 10.3390/s22134670.

Federico, S. Stroke prediction dataset. https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset, 2021.

Mohammed, R., Rawashdeh, J., and Abdullah, M. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In *2020 11th International Conference on Information and Communication Systems (ICICS)*, pp. Unknown, 2020. doi: 10.1109/ICICS49469.2020.239556. URL https://ieeexplore.ieee.org/document/9078901. Accessed: 2024-02-23.

Organization, W. H. The top 10 causes of death, 2020. URL https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. Accessed: 2024-02-23.

Quinlan, J. R. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

Satya, Akshit, Nitish, and Srinivas. Predicting stroke risk data cleaning. https://www.kaggle.com/code/satya514/predicting-stroke-risk-data-cleaning, 2021.

Wang, H., Yang, Y., Wang, H., and Chen, D. Soft-voting clustering ensemble. In Zhou, Z.-H., Roli, F., and Kittler, J. (eds.), *Multiple Classifier Systems*, pp. 307–318, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-38067-9.