STROKE PREDICTION

Cesario Laura Thoft, Ciciani Diego, Di Giammarco Susanna, Fazio Renato, Oddi Livia

Abstract

This project analyses stroke prediction. Our study focuses on predicting the possibility of an individual to have a stroke or not, based on clinical data from patients sourced from Kaggle.
After preprocessing the data, we tried to spot patterns in the data using three different machine learning algorithms: K-Nearest Neighbors (KNN), Logistic Regression, and Support Vector Machine (SVM), each of them has the objective to discover if someone can contract this medical condition or not, based on their life habit.
By applying the algorithms to the data, we observed suboptimal values in the resulting confusion matrices. This was probably due to the dataset's imbalance. To address the issue of the unbalanced dataset, we opted to employ the SMOTE (Synthetic Minority Over-sampling Technique) method.
The confusion matrices exhibited improvement after the application of SMOTE; however, it is noteworthy that the SMOTE technique radically alters the structure of the dataset by synthesizing additional minority class instances. An alternative approach that seemed promising was the adoption of model combination.
We compared the performance of individual models and combined models, applying them to both the SMOTE and non-SMOTE datasets. Our observations showed that, overall, the Support Vector Machine applied to the SMOTE training data was the best performer according to our requirements.

Introduction

In this study, we're focusing on predicting the likelihood of a stroke. Strokes happen when blood can't get to all parts of the brain, which can be deadly. They're a top cause of death globally, accounting for about 11% of all deaths according to the WHO. In 2019, strokes were the second most common cause of death worldwide.
Stroke is a major health concern, and efforts are ongoing to reduce its impact. One way to cut the risk is by making healthier lifestyle choices. Factors like high blood pressure, smoking, and diabetes can increase the risk of a stroke.
We're using a dataset from Kaggle, which contains clinical data from patients, building a model that can predict whether a person will have a stroke or not. By using the unique attributes of our dataset, we aim to improve our understanding of stroke risk and to find a good model for prediction.
The source code was retrieved from Requiem98's GitHub repository (1).

Related work

In recent years, various works employing Machine Learning algorithms for stroke prediction have been published. Govindarajan et al. utilized Artificial Neural Networks, Support Vector Machine, Decision Tree, Logistic Regression, and ensemble methods to classify stroke, achieving a notable accuracy of 95.3% with ANN. (2) Jeena and Kumar proposed a Support Vector Machine model with a linear kernel, attaining an accuracy of 91% for stroke prediction (3). Amini et al. performed research to predict stroke occurrence. They classified 50 risk variables for stroke, diabetes, cardiovascular disease, smoking, hyperlipidemia, and alcohol consumption in 807 healthy and unhealthy individuals using the K-nearest neighbor algorithm, achieving 94% accuracy (4).
Our study aims to enhance our understanding of stroke risk and develop a robust prediction model. Several studies have employed machine learning algorithms like Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Logistic Regression for stroke prediction, showing promising results. Therefore, our objective is to apply these algorithms sequentially on our dataset and see which one of them performs better to predict stroke.
By comparing the performance of these algorithms on our dataset, we hope to gain deeper insights into their relative effectiveness in predicting stroke occurrence.

Proposed method explained

To predict the likelihood of a person suffering from a stroke we tried to apply to the data the KNN algorithm, Logistic regression and Support Vector Machine, supervised learning algorithms trained on a labeled dataset.

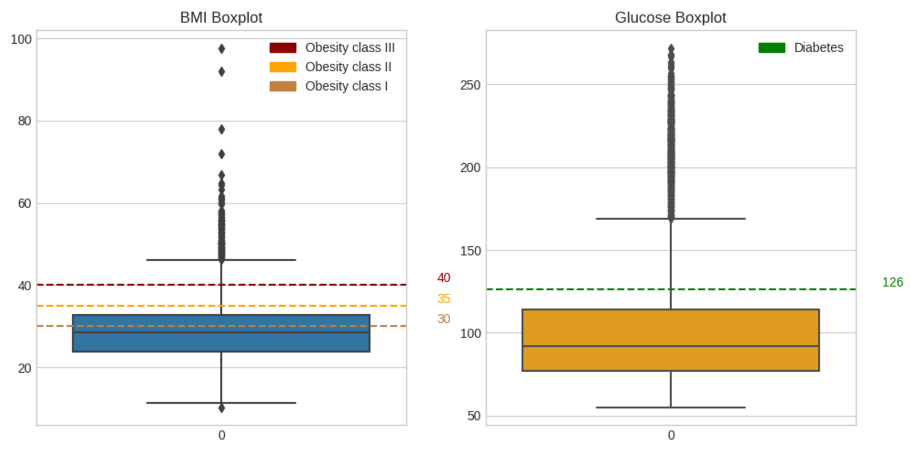
We decided to use these algorithms because they have the following strong points:
1. K-Nearest Neighbors (KNN)
The algorithm is simple and easy to understand. It can be used for classification or regression tasks and can be effective in high-dimensional spaces. However, the performance of KNN can be negatively impacted by changes in the data distribution and may not be suitable for large datasets due to its complexity in computation.
2. Logistic Regression
The strength of Logistic Regression lies in its ability to predict a binary outcome (0 or 1) from one or more input variables. Logistic regression can also handle data that is not linearly separable, thanks to its ability to work with the log-odds of the outcome. The model is robust and can provide a good starting point for prediction. However, logistic regression requires a careful choice of features and scaling of data, as it assumes linear relationships between input features and log-odds of the outcome.
3. Support Vector Machines (SVM)
One of the key strengths of SVM is its capability to locate optimal hyperplanes that maximize the margin between various classes. SVM is particularly effective at dealing with high-dimensional data. The decision function provided by SVMs is also particularly useful for multiclass classification. However, SVMs can be sensitive to the selection of hyperparameters (such as the choice of kernel and regularization parameter). They can also be slow to train for large datasets, especially when using complex kernels.

Additionally, we used a technique called **model combination** to try and improve the prediction of stroke occurrence.
We trained different models like Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) on our dataset. Each model has its own strengths and weaknesses, and combining them can enhance overall prediction performance.
The process starts with training the models, then generating estimates on the test data, and finally synthesizing those to create the final prediction. This approach, which often improves the prediction performance of the models, leverages the strengths of multiple models, especially when they are based on different assumptions or algorithms.

Dataset and benchmark

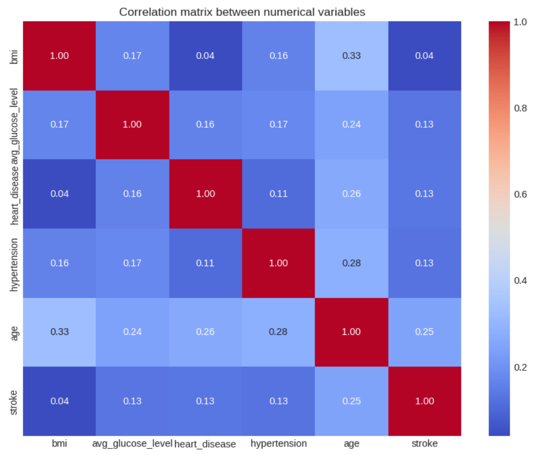
In our project the dataset (5) analyzed is taken from Kaggle and contains clinical informations (12 attributes) about 5110 patients. As a preliminar step we pre-processed the data:
1. In the *bmi* column we replaced the missing values with the mean;
2. In the *gender* column we removed the ”other” value, after making sure that it was only one observation over the whole dataset;
3. We checked for negative or missing values in the other columns.

Subsequentially we proceeded with a statistical descriptive analysis, with the aim of having a better understanding of the dataset and its variables. For example we used the boxplot to visualize the numerical variables:

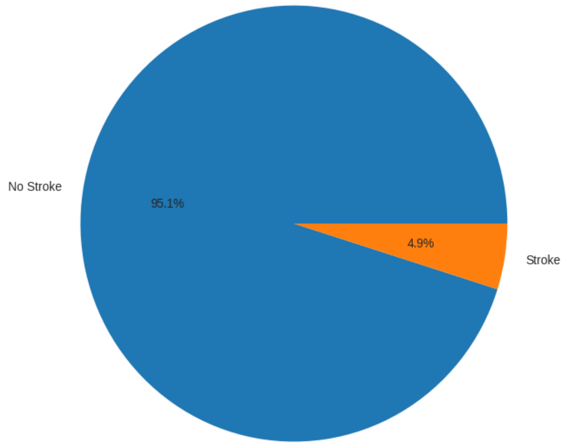


BMI and Glucose level boxplots

We also used a heatmap (Fig. a) to visualize the correlations between the numerical variables of our dataset, and a pieplot (Fig. b) to see the proportion of people who had or did not have a stroke:



(a) *Heatmap*



(b) *Stroke Pieplot*

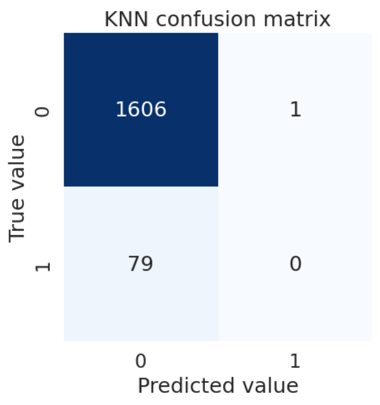
Before proceeding with the machine learning part of the analysis, we encoded the 'Gender', 'Ever Married', and 'Residence Type' variables using 0 and 1. The reason behind this is that many machine learning algorithms require numerical input, so we transformed these binary variables into a format that these algorithms can understand and use effectively.

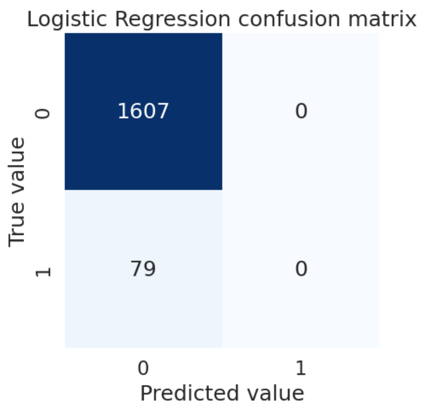
From the pieplot (Fig. b) we can observe that our dataset is indeed unbalanced: 95.1% of our subjects did not have a stroke, while just the 4.9% had it. Since this imbalance can lead to a model that is biased towards the majority class and performs poorly on the minority class, we tried to address the issue with the **SMOTE (Synthetic Minority Over-sampling Technique)** technique that is used to handle imbalanced datasets in machine learning.
Instead of simply duplicating existing instances from the minority class, SMOTE synthesizes new instances that are similar to the existing instances but not identical. This is done by selecting two or more similar instances from the minority class, drawing a line between them, and generating a new instance along this line.

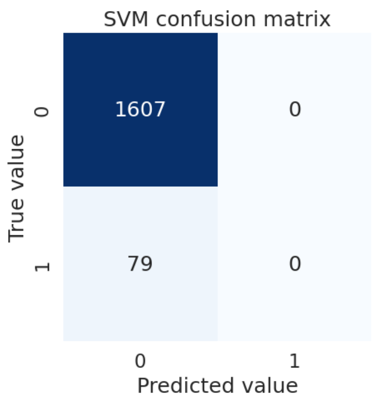
The dataset was divided into training and testing subsets, with the training subset comprising 67% of the data and the testing subset containing the remaining 33%. The models were trained using the training subset and then evaluated using the testing subset. The performance of these models was assessed using the *F1 score, recall, accuracy, and precision*, that are key metrics in machine learning for evaluating the performance of classification models. These metrics are based on the true positive, true negative, false positive, and false negative values derived from the model's predictions.

Experimental results

After applying the KNN, Logistic regression and Suppport Vector machine algorithms we obtained the following confusion matrices:

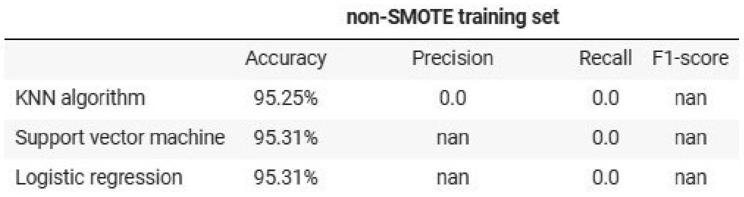






We can obivously see that the algorithms confusion matrices (almost) only predict 0's values. This is largely due to the fact that, as we observed in the statistical descriptive analysis, the dataset is unbalanced. As a result, the models struggle to correctly

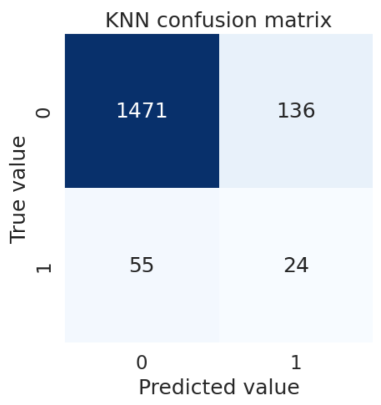
identifying the minority class (1's), resulting in the confusion matrices being dominated by 0's.
Also, if we look at the performance evaluation metrics, we have values only for the accuracy. We don't have values for the other performance evaluation metrics because of the way the confusion matrices came out:

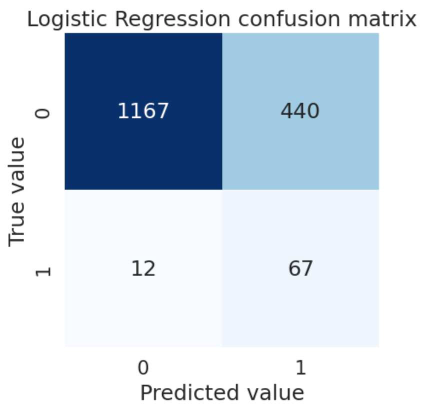


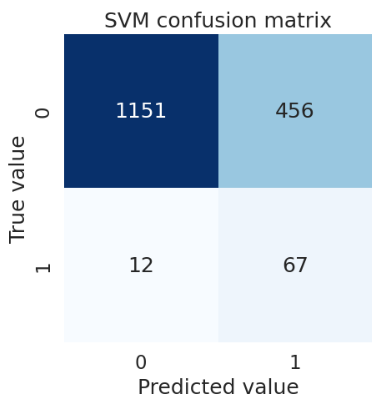
non-SMOTE training set

	Accuracy	Precision	Recall	F1-score
KNN algorithm	95.25%	0.0	0.0	nan
Support vector machine	95.31%	nan	0.0	nan
Logistic regression	95.31%	nan	0.0	nan

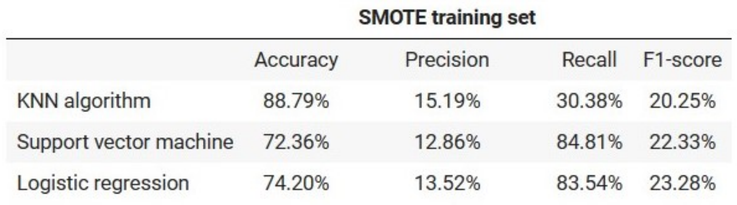
All these results suggest that more investigation is needed, and that's why we decided to use the SMOTE techinique to artifically balance the class distribution in the training set. We also thought about trying ensamble methods (model combination).
After applying the SMOTE to the training set the confusion matrices become better:







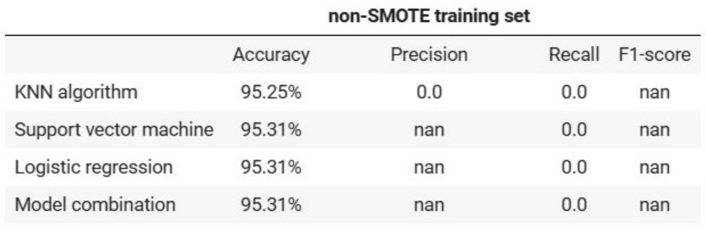
Also the performance evaluation metrics improved:



SMOTE training set

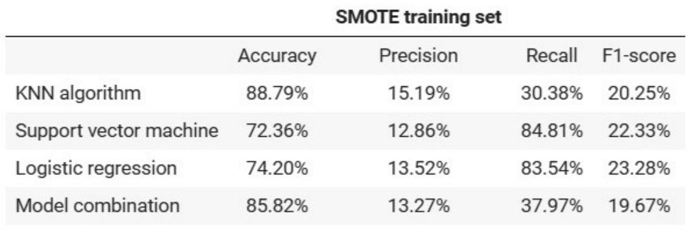
	Accuracy	Precision	Recall	F1-score
KNN algorithm	88.79%	15.19%	30.38%	20.25%
Support vector machine	72.36%	12.86%	84.81%	22.33%
Logistic regression	74.20%	13.52%	83.54%	23.28%

By trying the model combination on both the SMOTE and non-SMOTE training set we obtained the following results:



non-SMOTE training set

	Accuracy	Precision	Recall	F1-score
KNN algorithm	95.25%	0.0	0.0	nan
Support vector machine	95.31%	nan	0.0	nan
Logistic regression	95.31%	nan	0.0	nan
Model combination	95.31%	nan	0.0	nan



SMOTE training set

	Accuracy	Precision	Recall	F1-score
KNN algorithm	88.79%	15.19%	30.38%	20.25%
Support vector machine	72.36%	12.86%	84.81%	22.33%
Logistic regression	74.20%	13.52%	83.54%	23.28%
Model combination	85.82%	13.27%	37.97%	19.67%

Looking at the performance metrics above we can make some considerations about the employed models:
KNN Without SMOTE, the model achieves an accuracy of 95.25%, but the precision, recall, and F1-score are zero, indicating that the model is not performing well at identifying the positive class. With SMOTE, the accuracy drops to 88.79%, but the precision, recall, and F1-score increase, suggesting that the model is better at identifying the positive class.
Logistic Regression: Similar to KNN, the performance of Logistic Regression improves when SMOTE is applied. Without SMOTE, the accuracy is 95.31% but the precision, recall, and F1-score are NaN, suggesting that the model is struggling to identify the positive class. With SMOTE, the accuracy drops to 74.20%, but the precision, recall, and F1-score improve, indicating that the model is now better at identifying the positive class.

Support Vector Machine (SVM): The performance of SVM also improves when SMOTE is applied. Without SMOTE, the accuracy is 95.31% but the precision, recall, and F1-score are NaN, suggesting that the model is struggling to identify the positive class. With SMOTE, the accuracy drops to 72.36%, but the precision, recall, and F1-score improve, indicating that the model is now better at identifying the positive class.
Model Combination: The Model combination achieved an accuracy of 95.31% without SMOTE and 85.82% with SMOTE. This suggests that the combination of the models performs better than the individual models, especially when SMOTE is applied.
It appears that SMOTE is enhancing precision, recall, and F1-score, key metrics for model performance, especially with imbalanced datasets. While it slightly reduces model accuracy, it effectively improves the models' ability to identify the positive class, potentially introducing some noise or bias.

Conclusions and future works

In comparing the performance of various models applied to both the SMOTE and non-SMOTE training set, the Support Vector Machine (SVM) model demonstrated superior performance in predicting strokes. Despite having the second highest F1-score among the models, the SVM model stood out due to its balance between precision and recall, resulting in a low false positive rate. This balance is particularly important in medical diagnosis, where minimizing false positives can significantly reduce the risk of incorrect diagnoses.
While the accuracy of the SVM model was found to be lower than other models, the trade-off between accuracy and the false positive rate was deemed acceptable. This is because a model with a lower false positive rate but a lower accuracy can be seen as more conservative in its predictions, potentially reducing the risk of incorrect diagnoses.
In conclusion, the SVM model proved to be the most promising for predicting strokes, achieving a relatively high accuracy (72.36%) and the best recall score (84.81%) among the models tested. Its lower false negative rate compared to other models further strengthens its suitability for identifying individuals at risk of a stroke. Even though the SVM model's precision was slightly lower than other models, this was not as critical as preventing false negatives, which occur when a stroke goes undetected.
Given the obtained results we can say that, when dealing with unbalanced dataset for this type of classification problem, a good way to overcome it is to apply the SMOTE technique to the train set (Always being careful because it synthesize additional minority class instances). The SVM can be a good machine learning algorithm to predict if someone can have a stroke or not.

After trying to find a way to deal with unbalanced dataset by using both the SMOTE and the model combinations techniques, the next steps can be:
1. Trying *feature engineering* as an alternative to improve the model's performance;
2. *Exploring* more with *other models*, because while the Support Vector Machine (SVM) model has shown good results, there may be other machine learning algorithms that could perform better;
3. *Analyze the influence of* age, sex, and other *demographic factors* on the model's performance to provide a more comprehensive understanding of the underlying relationships between stroke risk and demographic features.

Work division

This study was conducted by a team of 5 students, and the work was divided among team members based on their expertise and skills.
Preprocessing: The task of data preprocessing and the code for statistical descriptive analysis was undertaken by Susanna and Renato.
Comments and Analysis: Laura and Livia were responsible for providing valuable comments and conducting the necessary analysis on the preprocessed data.
Training and Test Split: Diego handled the splitting of the dataset into training and testing subsets.
Machine Learning Algorithms: Laura and Diego took care of the implementation and configuration of machine learning algorithms (knn, Logistic Regression, SVM), and analysis of the results obtained.
Model Combination and Elbow Method: Livia executed these tasks to combine the individual models and identify the optimal number of clusters using the elbow method.go worked together to implement different machine learning algorithms such as Logistic Regression, SVM, and SMOTE.
Livia also provided extensive comments and corrections on confusion matrices, logistic regression, SVM, and SMOTE algorithm implementations. Additionally, she ensured proper references and citations in both the code and the report.

Laura and Livia contributed to the drafting and editing of the report, which comprehensively summarized the entire project's work, findings, and outcomes.
Diego, Renato, and Susanna performed thorough reviews and corrections of the report, ensuring accuracy and clarity in the presentation of the project's findings.

Bibliography

[1] Requiem98, "Fds final project," https://github.com/Requiem98/FDS-FinalProject, 2021.

[2] Govindarajan P., Soundarapandian R.K., and G. et al., "Classification of stroke disease using machine learning algorithms," *Neural Comput Applic*, 2019. [Online]. Available: https://link.springer.com/article/10.1007/s00521-019-04041-y

[3] R. S. Jeena and S. Kumar, "Stroke prediction using svm," *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 600–602, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:37534609

[4] Tazin T., Alam M. N., Dola N. N., Bari M. S., Bourouis S., and Monirujjaman Khan M., "Stroke disease detection and prediction using robust learning approaches," *Journal of healthcare engineering*, 2021. [Online]. Available: https://doi.org/10.1155/2021/7633381

[5] F. Soriano, "Stroke prediction dataset," 2021. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset