# Stroke Prediction in Patients with the help of Machine Learning Tools

COMP0172 - Coursework 1

Student number: 23127196
Candidate number: JQLW1

November 2023

## 1 Introduction

A stroke is a medical condition which requires immediate intervention. There exist two types of strokes: the first one is called "ischemic stroke" and it happens when the blood flow to an area of the brain is blocked (for example by a clot) and thus the cells in that area do not receive oxygen and nutrients anymore. This lack of blood damages the cells of the interested area. The second one is called "hemorrhagic stroke" and is caused by internal bleeding in an area of the brain. The leaked blood increases the pressure in that area damaging the respective cells. Strokes can cause long term disabilities and even death. An early diagnosis is essential in effectively treating this condition. Moreover, being able to know with a reasonable amount of confidence if a patient will be subject to a stroke allows the doctors to monitor his condition, taking preventive action where possible and intervening as soon as the problem arises [2]. Machine learning (ML) tools could be employed to achieve this objective. In particular, it could be built a predictive model that highlights patients who are likely to get a stroke. In this project, a dataset containing data from more than 5000 patients was used to try to create such a model.

## 2 Methods

The dataset considered contains data from 5110 patients. For every patient, twelve attributes were annotated. The attributes' meanings and values are summarised in Table 1.

In particular, the most relevant feature is the "stroke" attribute. This is the target feature of the model's predictions. As can be noticed by looking at the distribution of its values, reported in Fig. 2, and at the data reported in Table 1, our dataset is greatly unbalanced. Addressing this pronounced class imbalance was a crucial factor in selecting an appropriate evaluation metric for the model. Furthermore, by looking at the features' correlation matrix (reported

in Fig. 2), it can be noticed that no single attribute is directly correlated with the likelihood of experiencing a stroke. Consequently, more complex relations between the attributes must be considered in order to build a reliable solution. Furthermore, during the exploration phase, it was also investigated the presence of unknown or inconsistent values. In particular, it was found that 201 patients didn't have their BMI recorded and around 30% of the patient didn't have their smoking status recorded. It was also noticed the presence of an unknown value in the gender attribute of one patient.

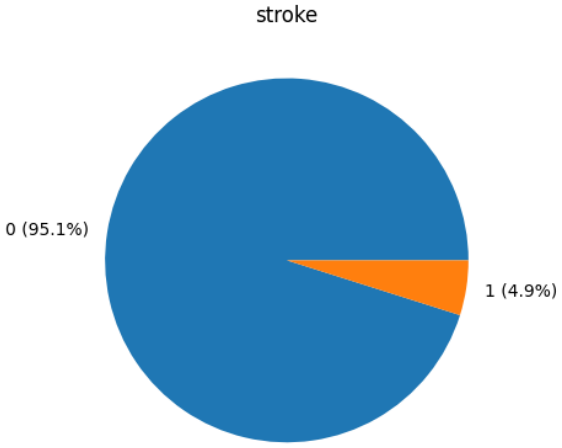



Figure 1: Class distribution in stroke features.

The dataset was then preprocessed in order to be used to train the ML models effectively. The first step was to handle the unknown values present in the dataset. For the BMI attribute, the missing values were substituted with the median BMI of the recorded patients. In particular, two values were used. The first one was calculated among the patients who didn't experience a stroke while the second one was computed considering who had a stroke. These values were then used to replace the missing ones of the patients based on whether they had a stroke. Conversely, the unknown values in the smoking status feature were not replaced with anything and, consequently, they were treated as a proper class of the attribute. As mentioned above, the gender attribute did also contain unknown values. Because this affected just one patient, who also didn't have a stroke, it was handled by just removing that patient from the dataset. Once the missing values were eliminated, most of the attributes' values were modified so that they were in a form more suited for training ML models. In particular, all the categorical features, that were not already encoded with only one and zero, have been encoded with the one-hot encoding. At the same time, all numerical attributes were scaled so that they had zero mean and unit variance.

Once the dataset was preprocessed, it was split into a train set and a test set. In particular, 30% of the dataset was held out for testing purposes. Because

| **Feature Name** | **Description** | **Possible Values** |
|---|---|---|
| Id | Unique identifier | Integers |
| gender | The gender of the patient | "Male" (41%)<br>"Female" (59%)<br>"Other" (0%) |
| age | Age of the patient | Float between 0.8 and 82 with a mean of 43 |
| hypertension | Whether or not the patient has hypertension | 0: no hypertension (90%)<br>1: has hypertension (10%) |
| heart_disease | Whether or not the patient has heart diseases | 0: no heart diseases (95%)<br>1: has heart diseases (5%) |
| ever_married | Whether the patient has ever been married | "No" (66%)<br>"Yes" (34%) |
| work_type | The type of work in which the patient is employed | "Children" (13%)<br>"Govt_job" (13%)<br>"Never_worked" (1%)<br>"Private" (57%)<br>"Self-employed" (16%) |
| Residence_type | The type of the patient's house | "Rural" (49%)<br>"Urban" (51%) |
| avg_glucose_level | Average glucose level in blood | Float between 55.12 and 271.74 with mean 106.15 |
| bmi | Body mass index | Float between 10.3 and 97.6 with mean 28.89 |
| smoking_status | The smoking history and status of the patient | "formerly smoked" (17%)<br>"never smoked" (37%)<br>"Smokes" (16%)<br>"Unknown" (30%) |
| stroke | Whether or not the patient had a stroke | 0: no stroke (95%)<br>1: had a stroke (5%) |

Table 1: Feature Description

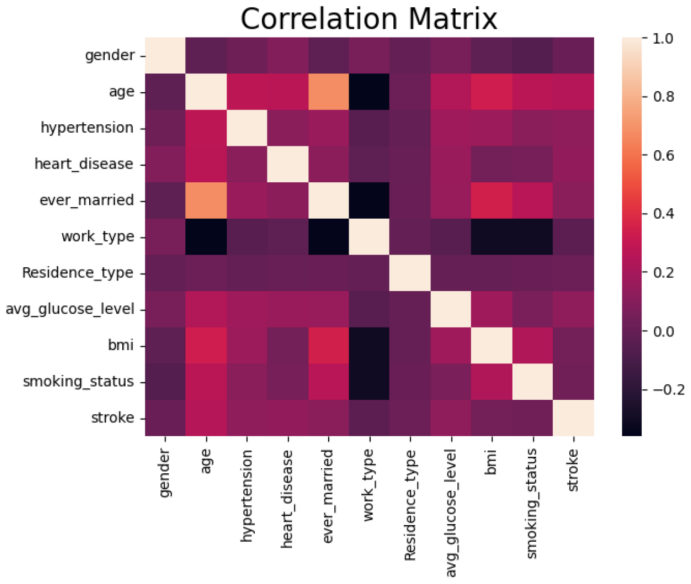



Figure 2: Attributes' correlation matrix

the dataset used had just 5000 thousand entries of tabular data, It was decided to focus solely on traditional machine learning models and not use any deep learning technique. In particular, four models were tested:

- K-NearestNeighbours (KNN)
- Logistic Regression
- Support Vector Machines (SVM)
- Random Forest

Every model was subjected to a hyperparameters tuning phase. Because the hyperparameter spaces of the first three models were small enough, it was possible to do an exhaustive search of them. Conversely, the hyperparameters space for the random forest had prohibitive dimensions. For this reason, it was explored using a random search of the space with 1000 iterations. The hyperparameters spaces tested are detailed in Table 2. Every set of hyperparameters was validated by doing a five-crossing validation on the train set. The metric chosen for the validation was the "f1" score. As already mentioned, this metric was chosen because of its capacity to compensate for the unbalanced nature of the dataset. Furthermore, other metrics were tested, such as the accuracy and the average precision, but they achieved unsatisfactory results.

| Name of the Parameter | Values |
|---|---|
| ***KNN*** | |
| n_neighbors | [1, 2, 3, 4, 5, 10, 15, ..., 1000] |
| weights | [”uniform”, ”distance”] |
| ***Logistic Regression*** | |
| penalty | [None,’l2’, ’l1’] |
| max_iter | [10, 50, 100, 150, 200, 250, 300, 500, 1000, 2500] |
| class_weight | [None, ’balanced’] |
| ***SVM*** | |
| kernel | ’poly’ |
| C | [ 0.4, 0.8, 1.2, ..., 10 ] |
| degree | [0, 1, 2, ..., 10] |
| ***Random Forest*** | |
| n_estimators | $\mathcal{U}([10, 11, 12, ..., 300])$ |
| max_depth | $\mathcal{U}([1, 2, 3, ..., 200])$ |
| min_samples_leaf | $\mathcal{U}([1, 2, 3, ..., 20])$ |
| min_samples_split | $\mathcal{U}([1, 2, 3, ..., 20])$ |

Table 2: Hyperparameter spaces searched during the hyperparameter tuning phase. The hyperparameter name reported in the table is the same one used in the respective scikit-learn implementation of the model. Note how the random forest Hyperparameter space was defined through random variables because it has been explored with a random search.

# 3 Results

During the tuning phase, it was noticed by looking at the validation scores (reported in Table 4) that SVM, KNN and Random Forest behaved similarly on this problem. On the other hand, logistic regression achieved considerably different results in this phase. Once the tuning phase was finished, the models were retrained on the whole train set using the best parameters found (detailed in Table 3) and then tested on the test set. By studying the models' performances on the test set, it was possible to confirm what was already noticed during the tuning phase. In particular, we had that SVM, KNN and Random Forest behaved similarly and they achieved considerably low f1 scores. On the other hand, logistic regression performed differently from the others and it achieved the highest f1 value, even though it was still far from optimal (Table 4). By studying the confusion matrices of the test phase (reported in Fig. 3), it was seen that the three models with low f1 scores did not manage to discriminate whether a patient would have a stroke. In particular, they just predicted no

stroke at all for most of the patients. Thus, they were characterised by extremely low recall values which led to low f1 values. Conversely, the logistic regression model was able to better discriminate whether a patient will have a stroke. In particular, it has the highest recall among all the models studied. Unfortunately, it also predicted a high number of false positives and thus it was characterised by a low precision. Furthermore, the ROC and the precision-recall curves of all the models were compared (Fig. 3). In particular, the areas under the curves were used as a metric of the models' performances. These metrics also confirmed that the logistic regression model performed better than the others. Finally, it is worth noticing how the models with the lowest f1 score also achieved the highest accuracy while the logistic regression achieved the lowest accuracy among all the models studied (Table 4).

| **Name of the Parameter** | **Values** |
|---|---|
| ***KNN*** | |
| n_neighbors | 1 |
| weights | "uniform" |
| ***Logistic Regression*** | |
| penalty | 'l1' |
| max_iter | 1 |
| class_weight | 'balanced' |
| ***SVM*** | |
| kernel | 'poly' |
| C | 8 |
| degree | 5 |
| ***Random Forest*** | |
| n_estimators | 15 |
| max_depth | 117 |
| min_samples_leaf | 1 |
| min_samples_split | 2 |

Table 3: Best parameter found during hyperparameter tuning

| **Model** | **F1 Score 5 Fold CV** | **F1 Score Test Set** | **Accuracy o Test Set** |
|---|---|---|---|
| KNN | $0.07 \pm 0.06$ | 0.06 | 91.85% |
| SVM | $0.12 \pm 0.04$ | 0.15 | 93.15% |
| Random Forest | $0.04 \pm 0.04$ | 0.11 | 94.91% |
| **Logistic Regression** | $\mathbf{0.23 \pm 0.02}$ | **0.23** | **73.26%** |

Table 4: Scores during validation and test phase for each model.

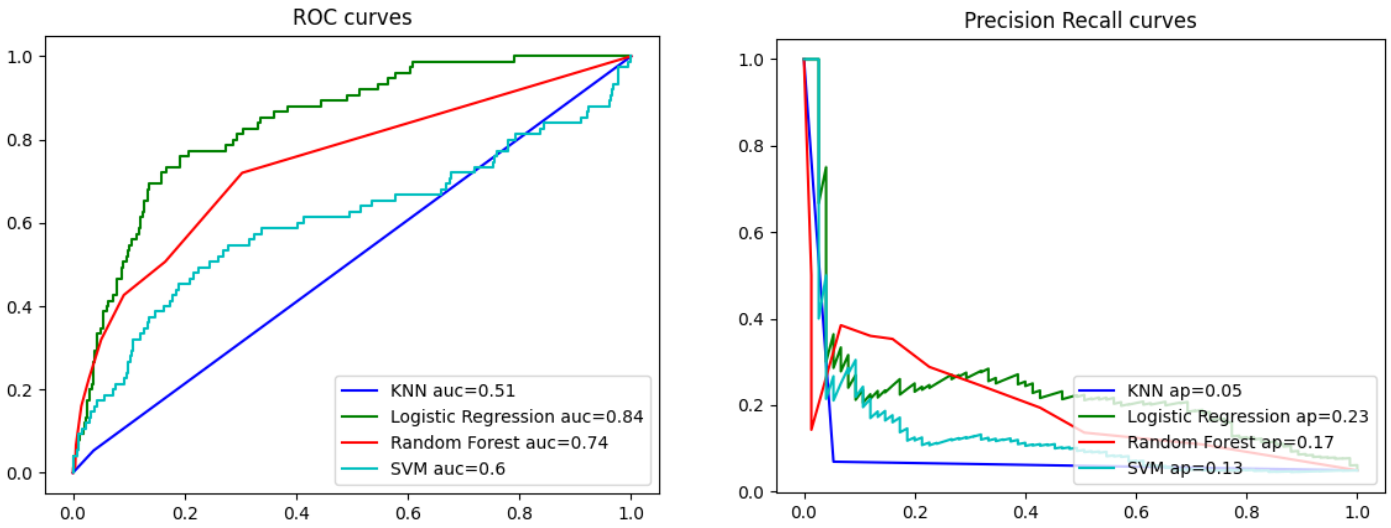
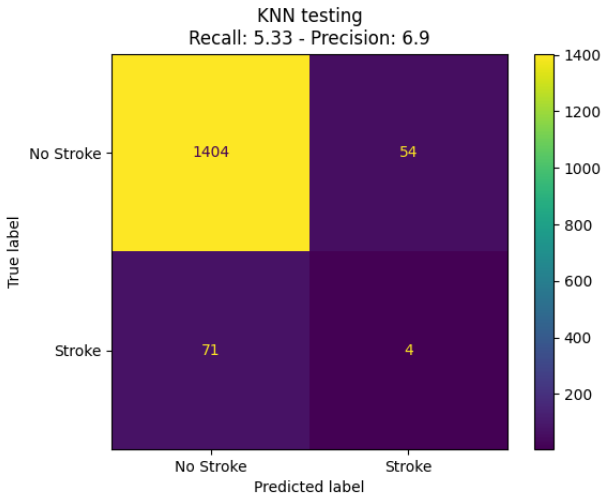
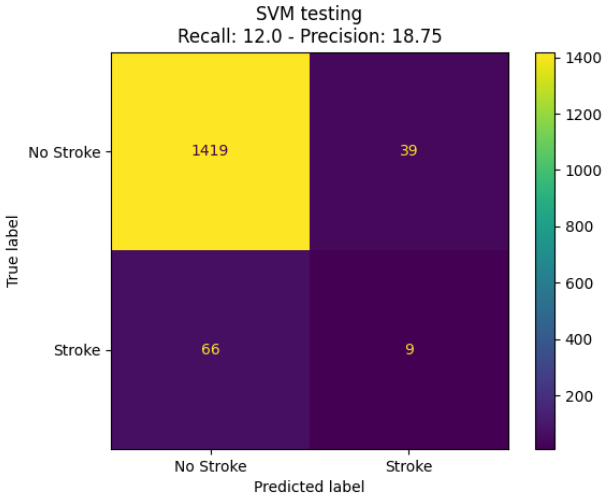


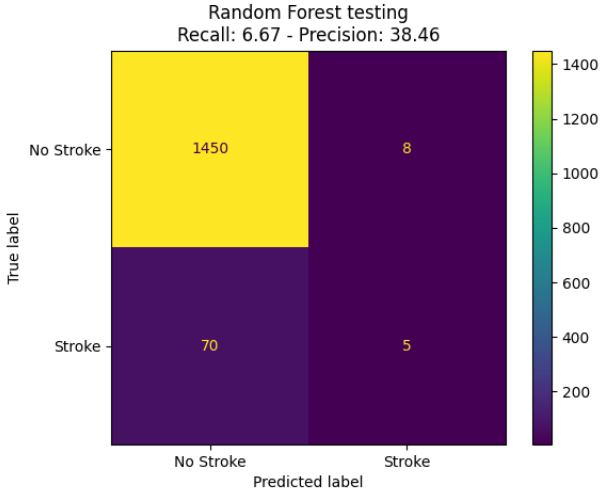

Figure 3: ROC and Precision Recall curves of all models.



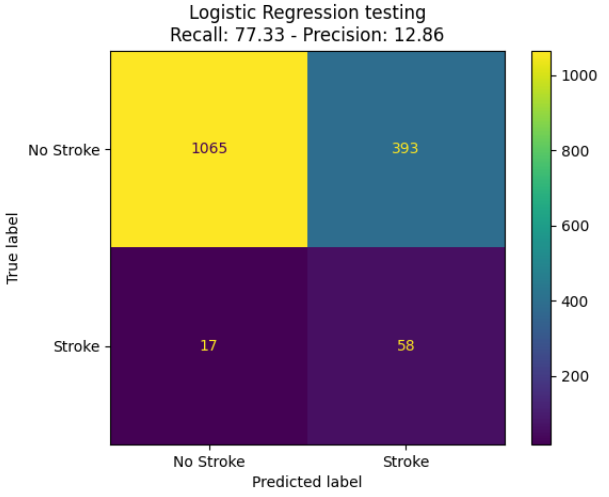

(a) KNN confusion matrix on test set




(b) SVM confusion matrix on test set




(c) Random forest confusion matrix on test set




(d) Logistic regression confusion matrix on test set

Figure 4: Models' confusion matrices on test set with relative Recall and precision

# 4 Discussion

From the results discussed in the previous section, it can be seen how SVM, KNN and Random Forest are not really useful for tackling this problem with the current implementation. Conversely, the trained logistic regression model could be useful in settings where it is possible to take some preventive action to help the patients highlighted by the model. Unfortunately, those actions must have no side effects (or at least they must not be too severe) because of the high number of false positive predictions generated. An example of such an action could be to prescribe an anticoagulant for a short period to prevent clots. Note how this solution has been proposed as an example and is not exempt from problems. In particular, it could be beneficial for ischemic strokes, which are the most common type of stroke. However, it could exacerbate the situation in case of hemorrhagic strokes. Further reasoning in this direction would be necessary.

Because the performances of the models tested were far from being optimal, there is probably room for several improvements. One approach that could be tried is to enrich the dataset. Because of the low incidence of stroke, very few patients in the dataset had one. Moreover, the data points with positive classes in the dataset are probably very sparse. This can be noticed by investigating the best set of parameters found for both the KNN model and random forest. In particular, the fact that the best number of neighbours found for the KNN model is one, and the best minimum number of data points for a leaf in the random forest model is also one, suggests that the data points with positive outcomes are very distant from each other. Consequently, the models didn't generalise well. With a larger dataset, we would have more cases and thus models with a lower generalisation error. Because the standard ML models failed to capture the relations between the feature of the dataset and the event of a stroke, another possible approach could be to use a deep learning model to tackle the problem. This would also be particularly useful if combined with the previous point due to the higher amount of data that is generally needed to train this type of model. Finally, all these models could be combined in an ensemble model in order to achieve higher performance than just the single models, as it was done in the work of Emon, Minhaz Uddin, et al [1].

# 5 Conclusion

In conclusion, the model developed could be beneficial if deployed in clinical settings only if are identified some preventive actions with few side effects. Moreover, there is ample room for improvement, and the previous section introduced some approaches to achieve that. Finally, it must be highlighted how the dataset does not report any data about its provenience (e.g. the ethnicity of the patient). Therefore, no evaluation of the fairness of the dataset, and thus of the trained model, can be done in advance. For this reason, if such a model would be deployed in a clinical setting, a close monitoring phase must be undertaken to

investigate how the model has been biassed and how this affects its predictions. Furthermore, continuous monitoring and evaluation of the model performance are advisable in general due to the many problems which characterise the ever changing environment of clinical settings. In particular, extra attention must be spent on monitoring if the model performance starts decreasing due to a distributional change between the training data and the new data. This can be particularly true if such a system is deployed in a clinical setting different from the one in which the data were collected in the first place, as it is in this case.

# References


[1] Minhaz Uddin Emon et al. "Performance Analysis of Machine Learning Approaches in Stroke Prediction". In: *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. Nov. 2020, pp. 1464–1469. DOI: 10.1109/ICECA49313.2020.9297525. URL: https://ieeexplore.ieee.org/abstract/document/9297525 (visited on 11/07/2023).

[2] *Stroke*. en. Section: conditions. Oct. 2017. URL: https://www.nhs.uk/conditions/stroke/ (visited on 11/25/2023).