STROKE PREDICTION MODEL

A24 [DA DE DS] Machine Learning with Python Labs
SPOC students project
Instructor: Hanna Abi Akl

Joanne ADAM
Data ScienceTech Institut
`joanne.adam@edu.dsti.institute`

0. Contents

1. Introduction

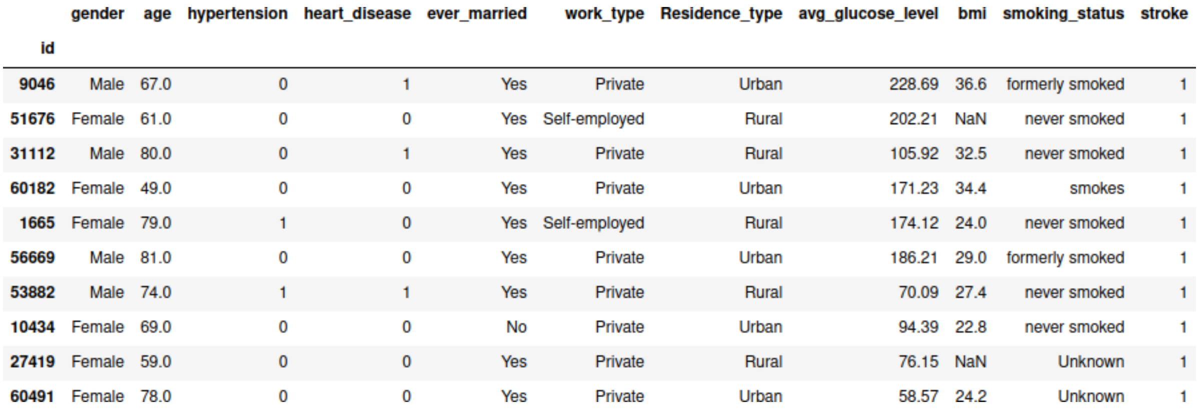
Brain stroke is a disease that lead to 11% of total deaths ans is the 2^{nd} cause of death according to the World Health Organisation (WHO). Early detection of stroke is critical in order to reduce the risk of death for patients. This study uses a dataset of personal and medical data for more than 5000 patients. Using machine learning approaches, this study will show how stroke could be diagnosed.

In a first part, the dataset will be presented. Secondly, feature selection and engineering is applied to calibrate the dataset in a better-suited form for machine learning algorithms. Thirdly the machine learning algorithms used in this study are introducted and the results presented to conclude with some interpretation of the results in the last part of this report.

2. Data Analysis

2.1. The dataset

The dataset consists of 5110 entries. Each entry correspond to one person's personal and medical information. A representation of the first entries of the dataset is presented in figure 1. The attributes, their possible values and frequencies are presented in tables 1 and 2.



	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
id											
9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smoked	1
31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smoked	1
56669	Male	81.0	0	0	Yes	Private	Urban	186.21	29.0	formerly smoked	1
53882	Male	74.0	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1
10434	Female	69.0	0	0	No	Private	Urban	94.39	22.8	never smoked	1
27419	Female	59.0	0	0	Yes	Private	Rural	76.15	NaN	Unknown	1
60491	Female	78.0	0	0	Yes	Private	Urban	58.57	24.2	Unknown	1

Figure 1: dataset head

id	Unique patient identifier
gender	Patient's gender
age	Age of the patient
hypertension	Whether or not the patient has hypertension
heart_disease	Whether or not the patient has a heart disease
ever_married	Whether or not the patient is married
work_type	Work status of the patient
Residence_type	Residence type of the patient
avg_glucose_level	Patient's average glucose level in the blood
bmi	Patient's body mass index
smoking_status	Patient's smoking status
stroke	Whether or not the patient had a stroke

Table 1: Dataset attributes

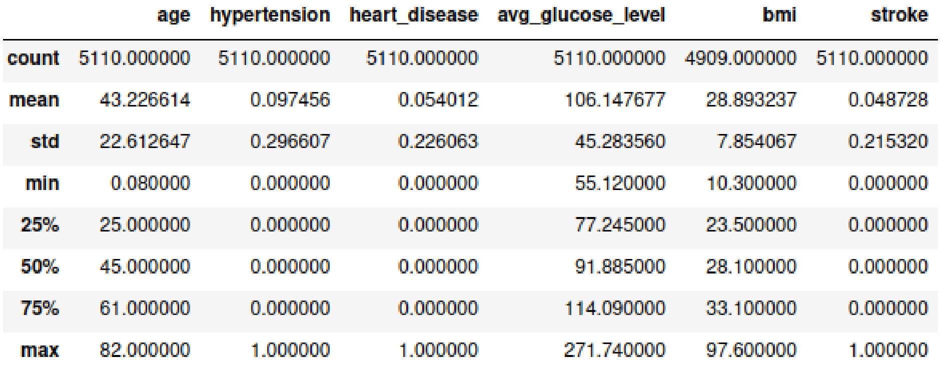
Attribute	**C/N**	**Value (frequency)**
gender	C	Male (2115), Female (2994), Other (1)
age	N	# (5110)
hypertension	C	0 (4612), 1 (498)
heart_desease	C	0 (4834), 1 (276)
ever_married	C	Yes (3353), No (1756)
work_type	C	Private (2925), Self-employed (819), Govt_job (657), children (687), Never_worked(22)
residence_type	C	Rural (2514), Urban (2596)
avg_glucose_level	N	# (5110)
bmi	N	# (4909)
smoking_status	C	formerly smoked (884), never smoked (1892), smokes (789), Unknown (1544)
stroke	C	0 (5861), 1 (249)

Table 2: Table of the attributes of the dataset with their categorical (C) or numerical (N) possible values.

2.2. Searching for outliers

The distribution of the numerical attributes enables the search of possible data outliers that could alter the analysis. Figures 2 and 3 present statistics and distribution of numerical attributes. The numerical attributes are of interest : 'age', 'avg_glucose_level' and 'bmi'.

The standard deviation of these attributes is always small compared to the mean which implies no presence of outlier values. The plot distribution confirms this observation (figure 3) for 'age' and 'avg_glucose_level' attributes. However, few values of BMI greater than 60 seems out of scale and are outliers. A body mass index of 60 is considered as hyper-obesity and is relatively rare but possible condition. These date could be kept as true data. Cases of hyper-hyper-obesity (bmi $>$ 80) are however extremely rare and could be errors in the measurements. Figure 4 presents the entries of the dataset where bmi is greater than 90.



	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000	5110.000000
mean	43.226614	0.097456	0.054012	106.147677	28.893237	0.048728
std	22.612647	0.296607	0.226063	45.283560	7.854067	0.215320
min	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	25.000000	0.000000	0.000000	77.245000	23.500000	0.000000
50%	45.000000	0.000000	0.000000	91.885000	28.100000	0.000000
75%	61.000000	0.000000	0.000000	114.090000	33.100000	0.000000
max	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

Figure 2: Statistics on numerical attributes of the dataset.

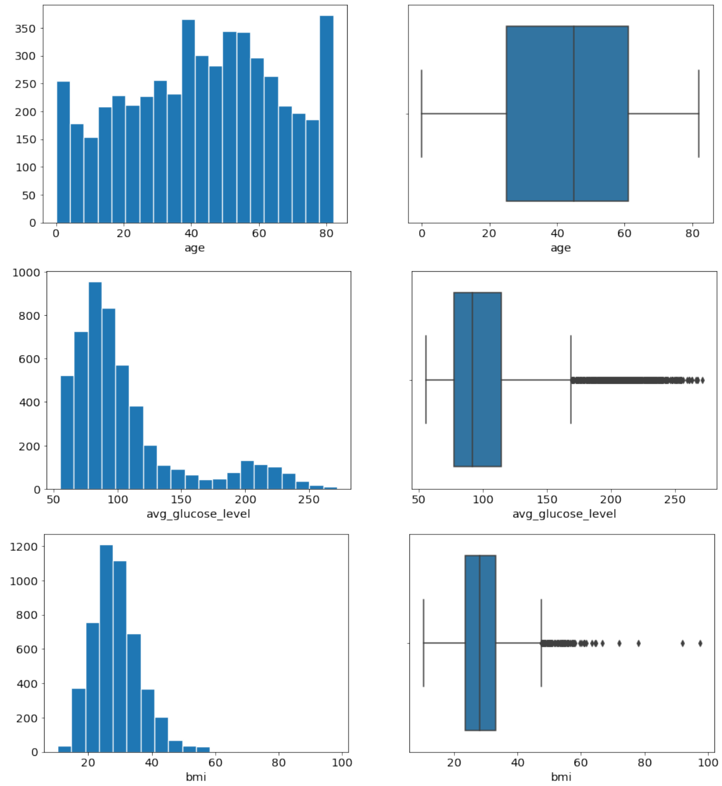
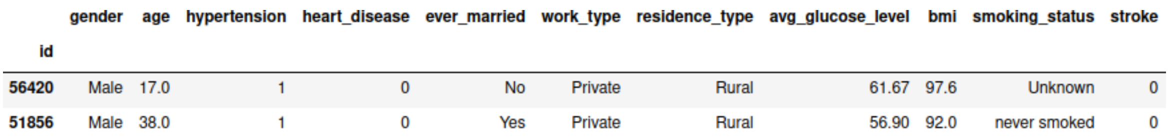


Figure 3: Distribution of the attributes ’age’ (first line), ’avg_glucose_level’ (second line) and BMI (third line). Histograms (left column) with bin widths of 50 and box plot (right column) showing the quartiles.



id	gender	age	hypertension	heart_disease	ever_married	work_type	residence_type	avg_glucose_level	bmi	smoking_status	stroke
56420	Male	17.0	1	0	No	Private	Rural	61.67	97.6	Unknown	0
51856	Male	38.0	1	0	Yes	Private	Rural	56.90	92.0	never smoked	0

Figure 4: Dataset entries where bmi is greater than 90.

2.3. Missing values

Figure 2 shows 201 missing values of bmi. The 1544 ’Unknown’ values of the ’smoking_status’ attribute as well as one ’Other’ value for the ’gender’ attribute can also be considered as missing values. Section 3 on feature engineering will prsent how these missing values are handled.

2.4. Class imbalance

Three attributes relating to heath issues (’stroke’, ’heart_disease’ and ’hypertension’) are bookean attributes. The figure 5 presents the distribution of these attributes and show a dataset imbalance problem. Indeed, the 0 values are well more represented than the 1 values.

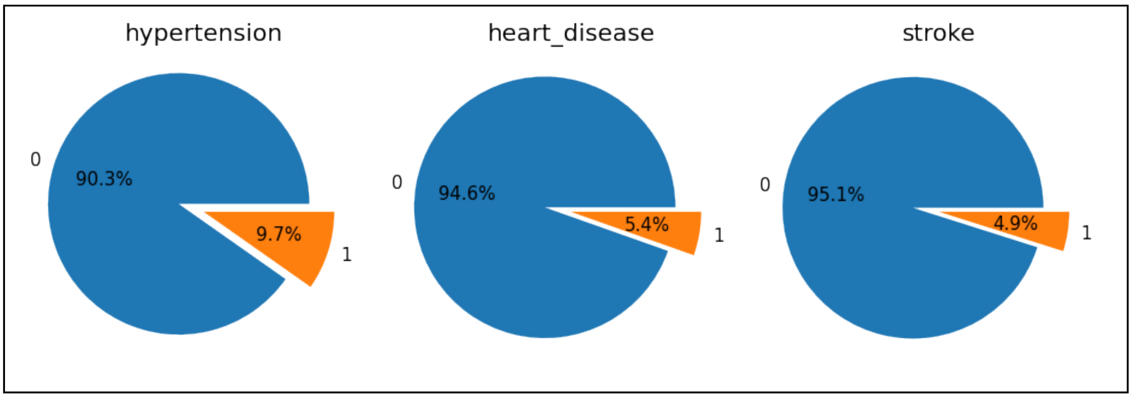


Figure 5: Distribution of the values of the attributes ’hypertension’, ’heart_disease’ and ’stroke’.

Because this study is about the predectability of one patient having a stroke, the health attributes could have high impact in the modeling. The correlation between these attributes (figure 6) seem however to suggest that they are only little correlated.

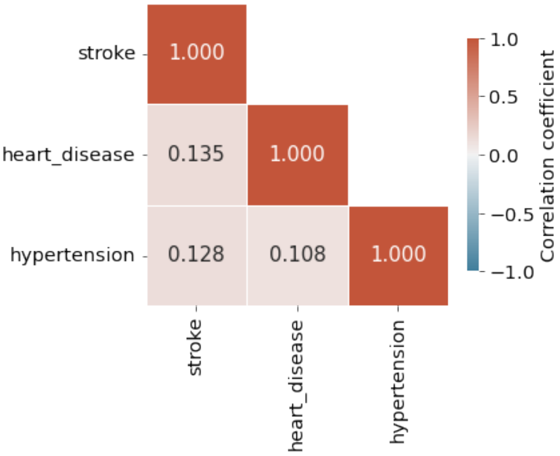


Figure 6: Correlation coefficients between the parameters ’stroke’, ’heart_disease’ and ’hypertension’.

2.5. Attributes name consistency

The attribute ’Residence_type’, starting with a capital letter, is renamed to ’residence_type’ so to have etymological consistency. Indeed, all attributes except ’Residence_type’ have names starting with a lower-case letter.

3. Feature engineering

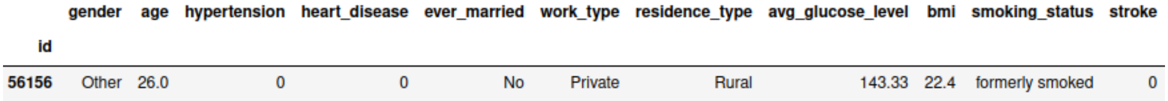
3.1. Categorical attributes to binary attributes

3.1.1 ’ever_married’ and ’residence_type’

Two attributes (’ever_married’ and ’residence_type’) have two and only two alternative values and can be represented using a boolean attribute.

3.1.2 ’gender’

The attribute ’gender’ can be represented as a boolean attribute however, one entry for ’gender’ in the dataset is ’Other’. The values for this entry are presented in figure 7. Because its ’stroke’ value is in the over-represented category, this entry is discarded.



	gender	age	hypertension	heart_disease	ever_married	work_type	residence_type	avg_glucose_level	bmi	smoking_status	stroke
id											
56156	Other	26.0	0	0	No	Private	Rural	143.33	22.4	formerly smoked	0

Figure 7: Entry of the database with a ’gender’ value of ’Other’.

3.1.3 'work_type'

Because the values of the attribute 'work_type' (*i.e.* 'Private', Govt_job', 'Never_worked', 'children' and 'Self-employed') are independant and unrelated, new attributes are created for each of the possible value. These new attribute are numerical, binary attributes.

3.1.4 'smoking_status'

The attribute 'smoking_status' has four possible values : 'unknown', 'never', 'formerly' and 'smokes'. In order to convert this categorical attribute to numerics, two options arise. The first option is ordinal encoding, by associating each value to a number (*e.g.* unknown → nan ; never → 1 ; formerly →2 ; smokes → 3). The second option is vector encoding, by creating new attributes for each possible value.

Trying ordinal encoding, the correlation matrix is computed for the whole dataset and presented in figure 8. No strong correlation is observed between the attribute 'smoking_status' and the other attributes.

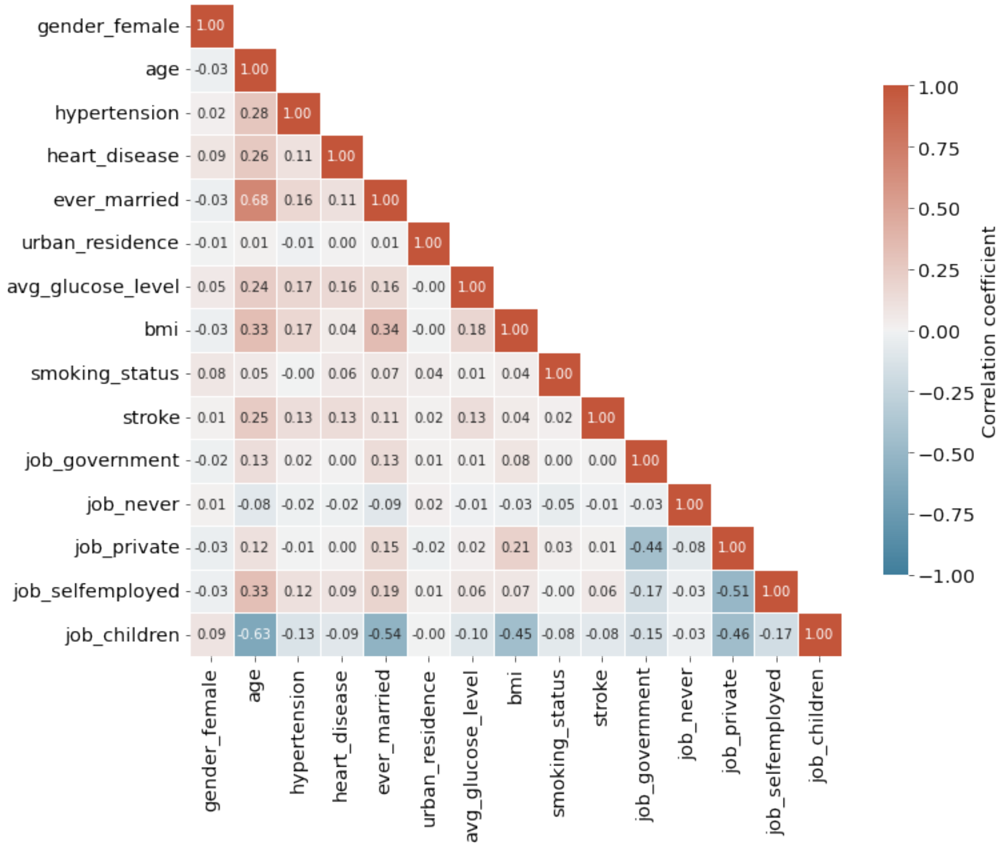


Figure 8: Correlation matrix for ordinal encoding of 'smoking_status' attribute.

Figure 9 presents the correlation matrix for the 'smoking_status' encoded with the vector encoding technique. Some relatively strong correlation appear between the new attributes and

’age’ or ’bmi’ for example as well as correlation between the new attribute themselves.

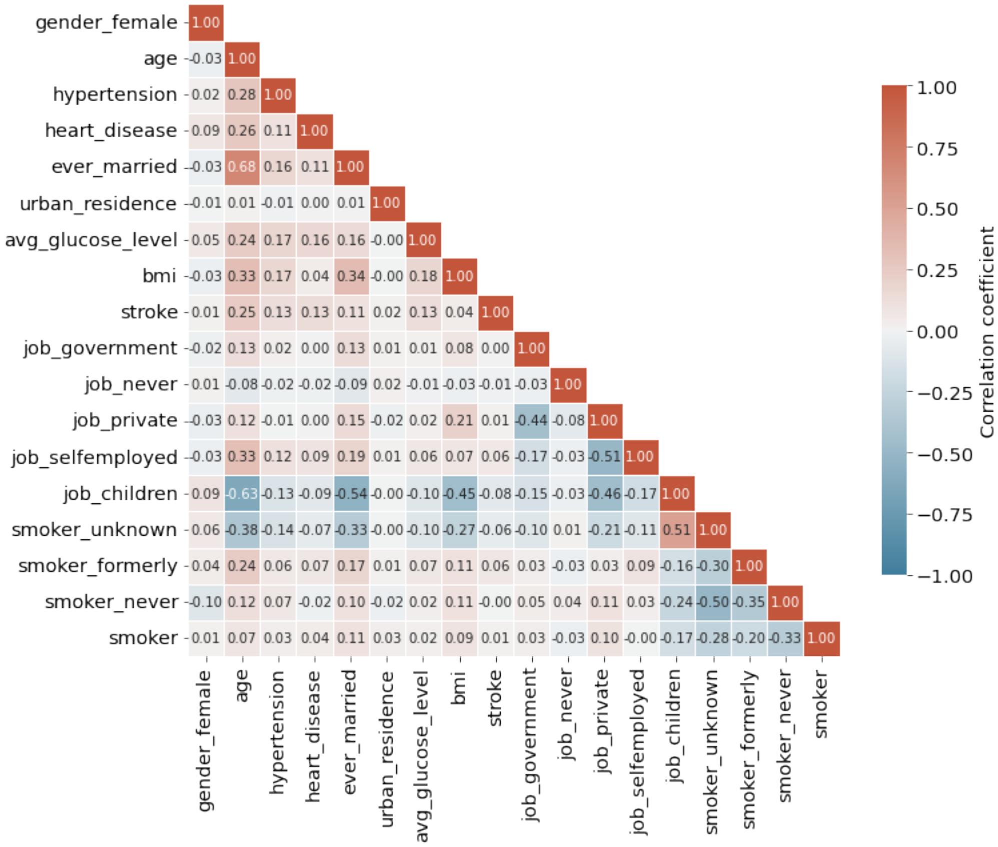


Figure 9: Correlation matrix for vector encoding of ’smoking_status’ attribute.

Because the dataset and the number of attribute are not large, vector encoding is retained. Moreover, this encoding brings the advantage of reducint biases in the data.

The attribute ’smoker_unknown’ has 1544 entries which correspond to the equivalent number of missing values. Further analysis would be necessary to assign values to the missing values however, considering the small correlation between this attribute and the ’stroke’ attribute, ’smoker_unknown’ is left like so in the dataset.

3.1.5 Summary of attribute changes

Modified attributes are summarized in table 3.

Old attribute name	**New attribute name**	**Old values**	**New values**
- ever_married	- ever_married	- No/Yes	0/1
- residence_type	- urban_residence	- urban/rural	0/1
- gender	- gender_female	- Male/Female	0/1
- work_type	- job_private - job_government - job_never - job_children - job_selfemployed	- Private/Govt_job/ Never_worked/children/ Self-employed	0/1 0/1 0/1 0/1 0/1
- smoking_status	- smoker_unknown - smoker_never - smoker_formerly - smoker	unkown/never/ formerly/smokes	0/1 0/1 0/1 0/1

Table 3: Categorical attributes transformed to binary attributes.

3.2. Missing values in the 'bmi' attribute

Figure 4 shows two unrealistic values of 'bmi' (body mass index greater than 90). Because the values of 'stroke' for these two entries is 0 which is over-represented, the two entries are removed from the dataset.

'bmi' attribute contains 201 missing values. A commun technique to fill the missing values is to assign each missing value with the average of the attribute over the dataset. In order to have a more accurate estimation of the missing values, 'bmi' values are compared to other numerical attributes with which the correlation is high. The 'age' attribute is a good candidate with a correlation coefficient of 0.33 (figure 9). The attribute 'job_children' would have been a better candidate with the highest correlation coefficient (0.48) however, its boolean encoding makes the operation difficult and not accurate. See figure 10 for the distribution of the attributes 'age' and 'bmi'.

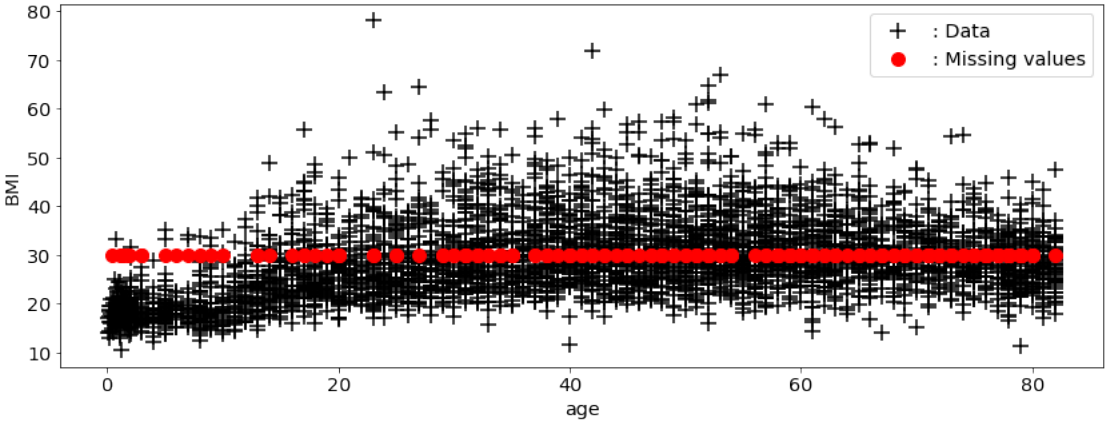


Figure 10: Distribution of the 'age' and 'bmi' attributes in black. Missing values of bmi are represented in red with a value of bmi of 30 for visual representation only.

Thanks to the discrete values of 'age' (values of 'age' are floats when lower than 2 and integers when greater or equal to 2), an average value of ’bmi’ is computed for each distinct ’age’ value. See figure 11 for estimated values of ’bmi’.

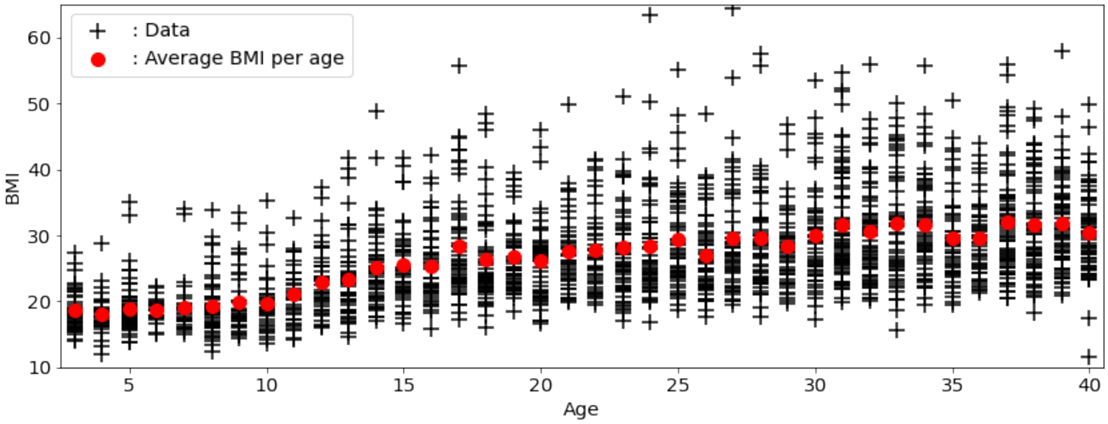


Figure 11: Distribution of the age and bmi (black) and average BMI over age to replace the missing values of bmi (red). Zoom between age 2 and 40 to better appreciate the variation in the data.

3.3. Ineffective attributes

Figure 9 shows that there is no correlation between ’urban_residence’ and any other attributes. This attribute is removed from the dataset.

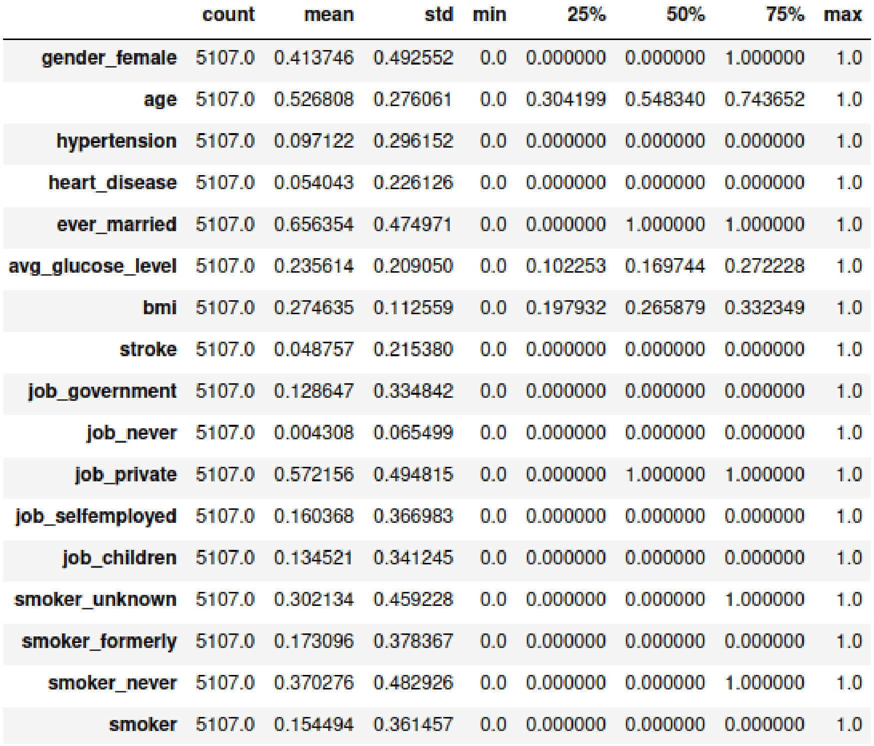
The attributes ’job_never’ and ’gender_female’ show very small correlation with other attributes. Their relevance might be discussed. Regarding the small size of the dataset, they are kept in this study.

3.4. Bounds of the numerical attributes

All the attributes of the dataset are numerical. All except three are boolean attributes and are in a the range: ’age’ $\in [0.08\ ;\ 82.0]$, ’bmi’ $\in [10.3\ ;\ 78.0]$ and ’avg_glucose_level $\in [106.16\ ;\ 271.7]$.

The values of these attributes are modified to be in the range $[0; 1]$. We thus obtain a more uniform dataset.

Figure 12 presents statistics of the final dataset.



	count	mean	std	min	25%	50%	75%	max
gender_female	5107.0	0.413746	0.492552	0.0	0.000000	0.000000	1.000000	1.0
age	5107.0	0.526808	0.276061	0.0	0.304199	0.548340	0.743652	1.0
hypertension	5107.0	0.097122	0.296152	0.0	0.000000	0.000000	0.000000	1.0
heart_disease	5107.0	0.054043	0.226126	0.0	0.000000	0.000000	0.000000	1.0
ever_married	5107.0	0.656354	0.474971	0.0	0.000000	1.000000	1.000000	1.0
avg_glucose_level	5107.0	0.235614	0.209050	0.0	0.102253	0.169744	0.272228	1.0
bmi	5107.0	0.274635	0.112559	0.0	0.197932	0.265879	0.332349	1.0
stroke	5107.0	0.048757	0.215380	0.0	0.000000	0.000000	0.000000	1.0
job_government	5107.0	0.128647	0.334842	0.0	0.000000	0.000000	0.000000	1.0
job_never	5107.0	0.004308	0.065499	0.0	0.000000	0.000000	0.000000	1.0
job_private	5107.0	0.572156	0.494815	0.0	0.000000	1.000000	1.000000	1.0
job_selfemployed	5107.0	0.160368	0.366983	0.0	0.000000	0.000000	0.000000	1.0
job_children	5107.0	0.134521	0.341245	0.0	0.000000	0.000000	0.000000	1.0
smoker_unknown	5107.0	0.302134	0.459228	0.0	0.000000	0.000000	1.000000	1.0
smoker_formerly	5107.0	0.173096	0.378367	0.0	0.000000	0.000000	0.000000	1.0
smoker_never	5107.0	0.370276	0.482926	0.0	0.000000	0.000000	1.000000	1.0
smoker	5107.0	0.154494	0.361457	0.0	0.000000	0.000000	0.000000	1.0

Figure 12: Statistics on dataset after feature engineering.

4. Modeling

4.1. Dataset splitting

The goal of this study is to predict whether a patient is likely to get a stroke. The target solution is the 'stroke' attribute while the data are all the remaining attributes.

80% of the dataset is used to train the algorithms and the remaining 20% are used to test the models. K-fold cross-validation is performed to reduce variance and incertainties due to random selection of the training and testing datasets. Results are averaged on five folds.

4.2. Class imbalance

Class imbalance on the target attribute 'stroke' (see section 2.4) could lead to failling model training and predictions. Oversampling might be necessary to counterbalance this problem. Figure 13 presents the distribution of the attribute 'stroke' as a function of 'age', 'avg_glucose_level' and 'bmi' (*i.e.* the three numerical values from the dataset).

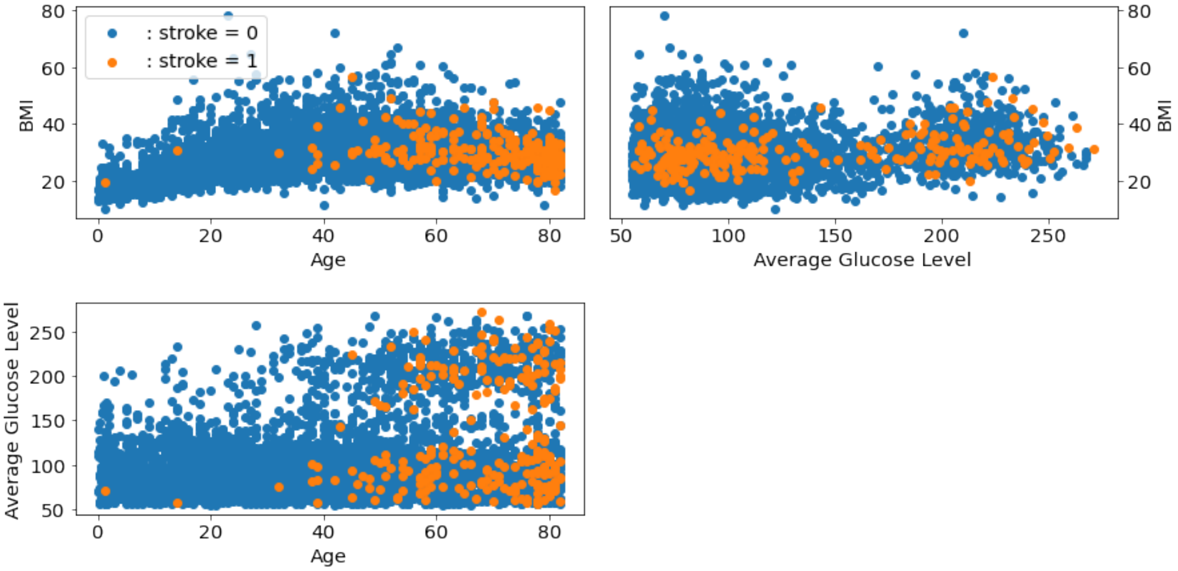


Figure 13: Distribution of the 'stroke' values as a function of attributes 'age', 'bmi' and 'avg_glucose_level'.

Class imbalance is handled using the Synthetic Minority Oversampling Technique (SMOTE). Figures 14 and 15 present the distribution of the over-sampled 'stroke' attribute and correlation matrix respectively. As a comparaison with the correlation matrix before over-sampling (figure 9), only small alteration is observed.

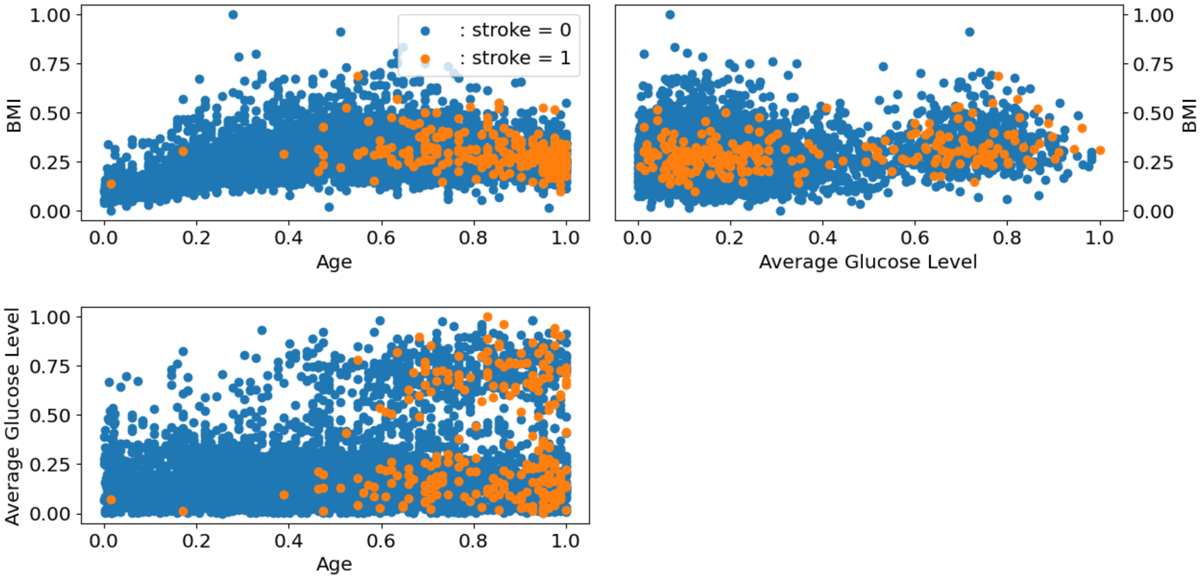


Figure 14: Distribution of the 'stroke' values as a function of attributes 'age', 'bmi' and 'avg_glucose_level' with oversampling of the minority attributes.

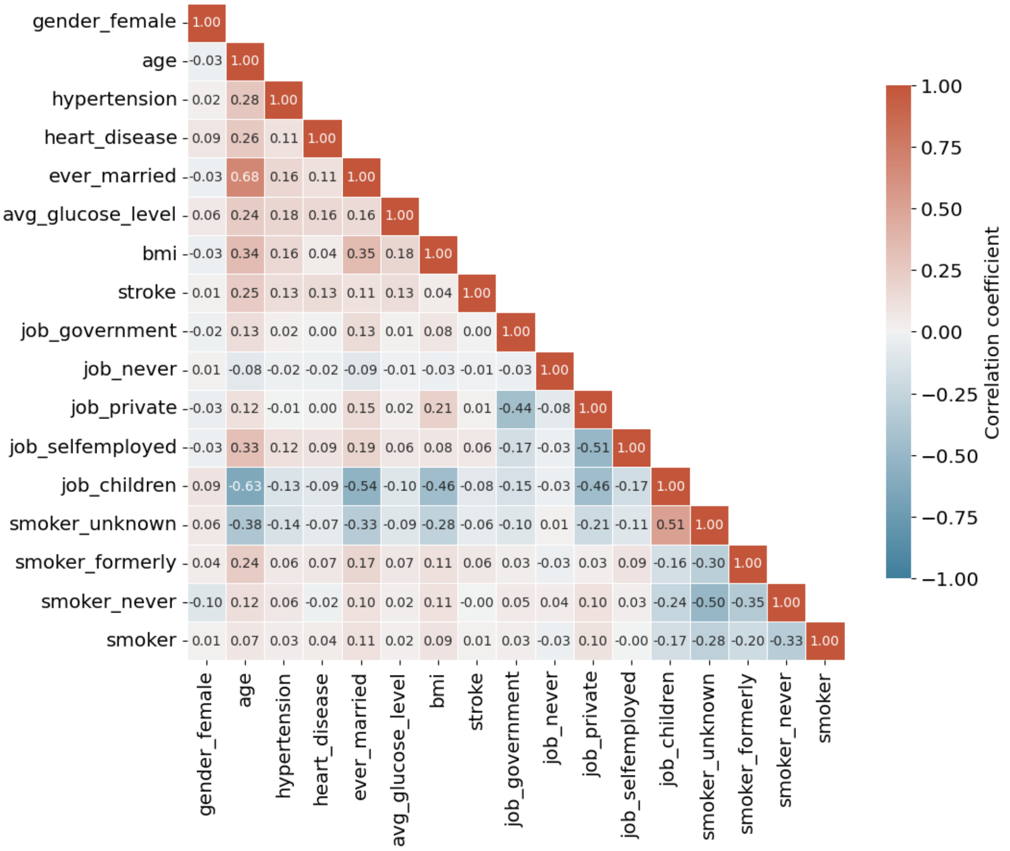


Figure 15: Correlation matrix with oversampling of the minority attributes.

Attribute	Original dataset	Oversampled dataset
age	#	#
avg_glucose_level	#	#
bmi	#	#
gender_female	0 (58%), 1 (41%)	0 (60%), 1 (39%)
hypertension	0 (90%), 1 (9%)	0 (88%), 1 (12%)
heart_disease	0 (94%), 1 (5%)	0 (91%), 1 (9%)
ever_married	0 (34%), 1 (65%)	0 (25%), 1 (75%)
stroke	0 (95%), 1 (4%)	0 (55%), 1 (45%)
job_government	0 (87%), 1 (12%)	0 (88%), 1 (12%)
job_never	0 (99%), 1 (0%)	0 (99%), 1 (1%)
job_private	0 (42%), 1 (57%)	0 (40%), 1 (60%)
job_selfemployed	0 (83%), 1 (16%)	0 (80%), 1 (20%)
job_children	0 (86%), 1 (13%)	0 (92%), 1 (8%)
smoker_unknown	0 (69%), 1 (30%)	0 (74%), 1 (26%)
smoker_formerly	0 (82%), 1 (17%)	0 (77%), 1 (23%)
smoker_never	0 (62%), 1 (37%)	0 (63%), 1 (37%)
smoker	0 (84%), 1 (15%)	0 (84%), 1 (16%)

Table 4: Table of attributes of the datasets with their possible values and corresponding frequency. The original dataset has 5107 entries and the oversampled dataset has 8802 entries.

4.3. Model training and evaluation using classification algorithms

Classification algorithms are applied to the dataset to predict the value of the 'stroke'. The classifiers used are Logistic Regression, Random Forest and Multi-layer Perceptron. Results for each of the algorithms are presented below.

Several evaluation metrics are computed for each modeling and for the dataset with or without oversampling :

- **Recall** $\in [0;1]$: measures how much of the positive data are well recovered from the dataset.
- **Precision** $\in [0;1]$: measures how much of the positive prediction are correct,
- **F1-score** $\in [0;1]$: informs on the performance of the prediction. Maximizing F1-score implies maximizing both the recall and precision.
- **Accuracy** $\in [0;1]$: informs on the ability of the model to make a correct prediction.

4.3.1 Logistic Regression

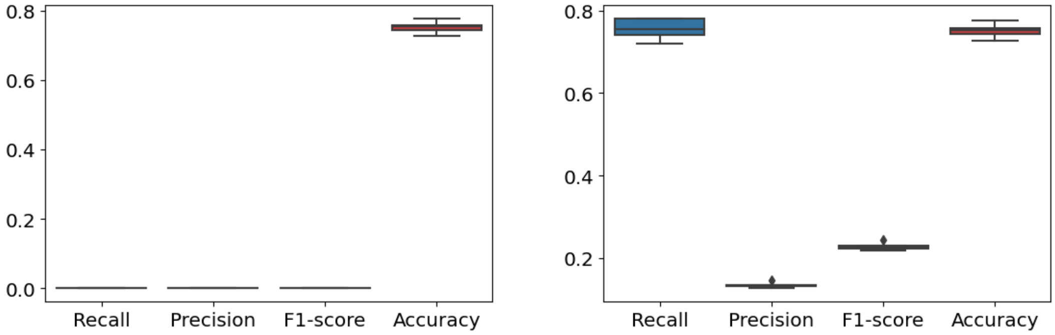
Logistic regression algorithm from the Python `sklearn` library search for a model that minimizes the residual squared sum between the observations and the model using linear approximation.

Metrics presented in Table 5 and figure 16 show that oversampling helps improve the identification of the positive target (higher values of recall). Although some improvement is observed, applying oversampling to the data does not improve the performance of the prediction (*i.e.* precision and F1-score values remain small). Results are consistent along each set of five runs.

The process time on a regular laptop is 0.074 s for the original dataset sampling and 0.234 s for the oversampled dataset.

	Run	Recall	Precision	F1-score	Accuracy
No SMOTE	1	0.0	0.0	0.0	0.756
	2	0.0	0.0	0.0	0.753
	3	0.0	0.0	0.0	0.727
	4	0.0	0.0	0.0	0.743
	5	0.0	0.0	0.0	0.776
SMOTE	1	0.74	0.136	0.229	0.756
	2	0.72	0.131	0.222	0.753
	3	0.78	0.137	0.218	0.726
	4	0.78	0.134	0.229	0.743
	5	0.76	0.146	0.244	0.775

Table 5: Recall, precision, F1-score and accuracy metrics computed for five folds using the original dataset sampling (no SMOTE) and the oversampled dataset (SMOTE) and logistic regression algorithm.



(a) No SMOTE

(b) SMOTE

Figure 16: Boxplot representation of the recall, precision, F1-score and accuracy metrics computed for five folds using the original dataset sampling (a) and the oversampled dataset (b) and logistic regression algorithm.

4.3.2 Random forest

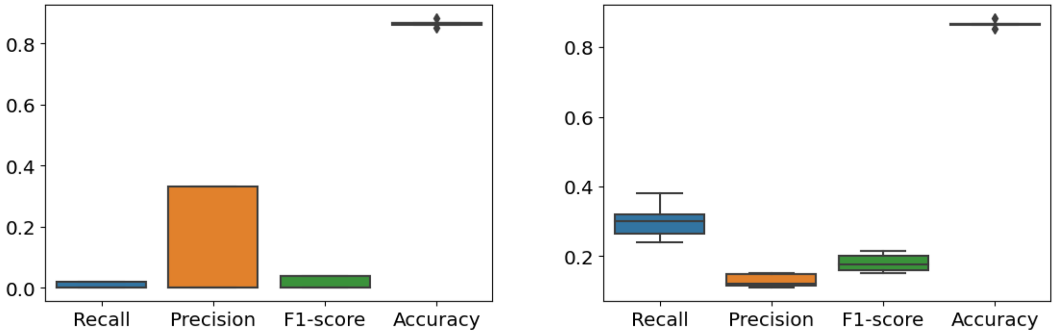
Random Forest algorithm from the Python `sklearn` library search for a best-fitting model using decision tree classifiers on sub-samples of the dataset.

Metrics presented in Table 6 and figure 17 show that oversampling slightly improves the identification of the positive target (higher values of recall). Although some improvement is observed, applying oversampling to the data does not improve the performance of the models (*i.e.* recall, precision and F1-score values remain small). Results are consistent along each set of five runs.

The process time on a regular laptop is 0.593 s for the original dataset sampling and 1.069 s for the oversampled dataset.

	Run	Recall	Precision	F1-score	Accuracy
No SMOTE	1	0.0	0.0	0.0	0.867
	2	0.02	0.33	0.04	0.883
	3	0.0	0.0	0.0	0.852
	4	0.0	0.0	0.0	0.863
	5	0.02	0.33	0.04	0.867
SMOTE	1	0.240	0.109	0.150	0.867
	2	0.300	0.150	0.200	0.883
	3	0.320	0.120	0.174	0.852
	4	0.380	0.148	0.213	0.863
	5	0.265	0.115	0.160	0.867

Table 6: Recall, precision, F1-score and accuracy metrics computed for five folds using the original dataset sampling (no SMOTE) and the oversampled dataset (SMOTE) and random forest algorithm.



(a) No SMOTE

(b) SMOTE

Figure 17: Boxplot representation of the recall, precision, F1-score and accuracy metrics computed for five folds using the original dataset sampling (a) and the oversampled dataset (b) and random forest algorithm.

4.3.3 Multi-layer Perceptron

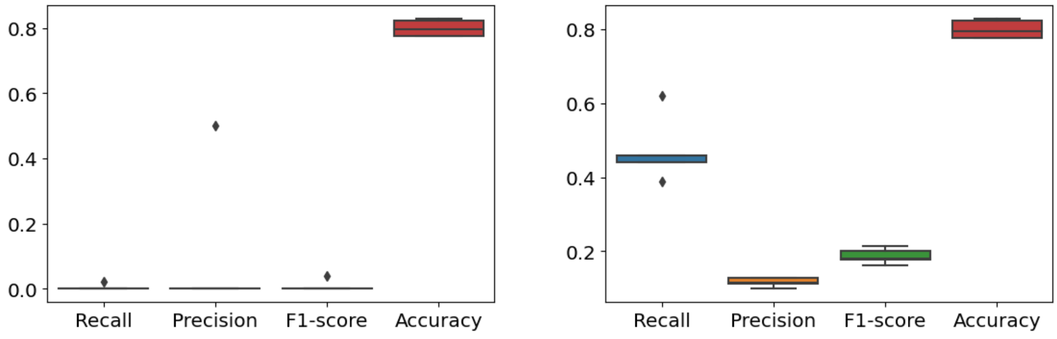
Multi-layer Perceptron algorithm from the Python `sklearn` library search for a best-fitting model using non-linear neural networks.

Metrics presented in Table 7 and figure 18 show that oversampling improves the identification of the positive target (higher values of recall). Although some improvement is observed, applying oversampling to the data does not improve the performance of the models (*i.e.* recall, precision and F1-score values remain small). Results are consistent along each set of five runs.

The process time on a regular laptop is 5.871 s for the original dataset sampling and 39.602 s for the oversampled dataset.

	Run	Recall	Precision	F1-score	Accuracy
No SMOTE	1	0.0	0.0	0.0	0.795
	2	0.0	0.0	0.0	0.822
	3	0.0	0.0	0.0	0.777
	4	0.0	0.0	0.0	0.777
	5	0.02	0.5	0.04	0.828
SMOTE	1	0.460	0.112	0.180	0.795
	2	0.460	0.129	0.202	0.882
	3	0.440	0.099	0.162	0.777
	4	0.620	0.129	0.214	0.777
	5	0.388	0.115	0.178	0.828

Table 7: Recall, precision, F1-score and accuracy metrics computed for five folds using the original dataset sampling (no SMOTE) and the oversampled dataset (SMOTE) and multi-layer perceptron algorithm.



(a) No SMOTE

(b) SMOTE

Figure 18: Boxplot representation of the recall, precision, F1-score and accuracy metrics computed for five folds using the original dataset sampling (a) and the oversampled dataset (b) and multi-layer perceptron algorithm.

5. Interpretation

Figure 19 summarizes the average metrics obtained for each algorithm. While all methods show similar accuracy, the logistic regression algorithm show better results of recall scores using oversampled dataset with very small computation time.

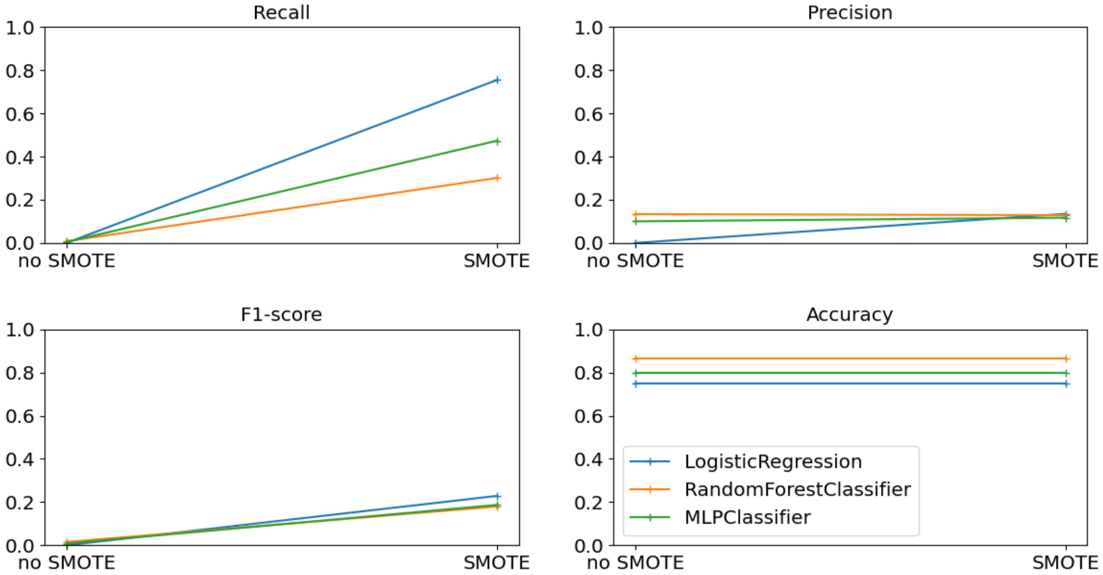


Figure 19: Average Recall, Precision, F1-score and Accuracy scores computed for Logistic Regression, Random Forest and Multi-Layer Perceptron algorithms using either original dataset sampling (no SMOTE) or oversampled dataset (SMOTE).

With an average recall score of 0.76, linear regression models are able to confidently predict true positive values of stroke. In a medical sudy like this study, one would expect no less than robust prediction of true positives.

Prediction scores are always below 0.15 which show that all the models predict a large quantiy of false positives. SMOTE application to counteract the class imbalance is insufficient.

With F1-score values lower than 0.3, the models performance is considered as poor.

The accuracy of the models average around 0.8. While this score seems high, in a medical study such accuracy might not be sufficient and further analysis or additional dataset inquiry might be necessary.

Out of curiosity, the algorithms have been applied using my personal and medical data. None of the models predicts me a risk of stroke, regarding my personal data. My grand-father died of stroke recently. The logistic regression and multi-layer perceptron models whould have predicted him a risk of stroke while the random forest model would not predict him any risk of stroke.

6. Conclusion

This study used a dataset of personal and medical data of more than 5000 patients in order to predict the event of stroke. After data analysis and feature selection, three machine learning models have been trained and tested. Results show that a linear approach is sufficient in order to accuratly predict a stroke event for a patient. The model do however predicts a high rate of false positives wich will lead to further clinical testing to patients that actually are not at risk of stroke. In medical studies, such prediction is best.