MACHINE LEARNING WITH PYTHON

A REPORT ON A STROKES DATA SET

30 DE ABRIL DE 2024

MATÍAS BOTTARINI
Data Science Student at DSTI

Goal of the document

The goal of the following document is to shortly resume the results and the approach regarding when creating a Machine Learning Pipeline on the Strokes Data Set.

The report is a resume of the jupyter notebook called *strokes_ml.ipynb* found in the folder *notebook* of the provided git repository. In the notebook you will find comments before each piece of code.

For this reason we are going to divide the report in 4 parts:

- Data analysis
- Feature selection
- Model Training
- Model Evaluation

Data Analysis

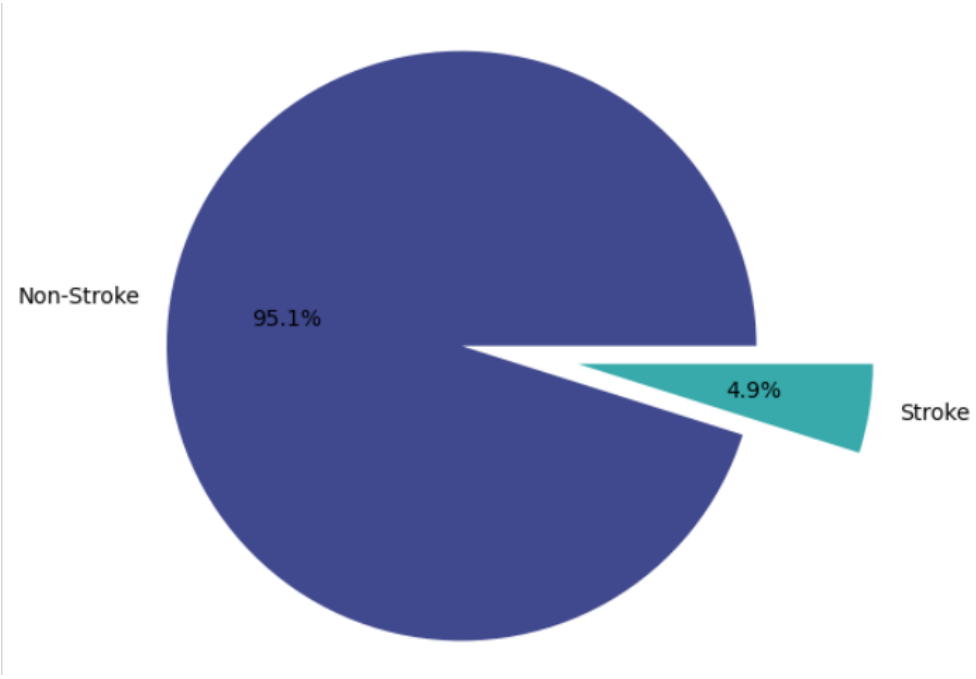
We observe **4 discrete variables** (int64) with our target variable **stroke** inside them, **3 continuous variables** (float64) and **5 variables** that as labels or factors (object)

From the 12 variables, we can say that we will have 11 explanatory variables and 1 output variable (**stroke**)

We have found NA values only on the **bmi** variable, the rest of the variables are filled.

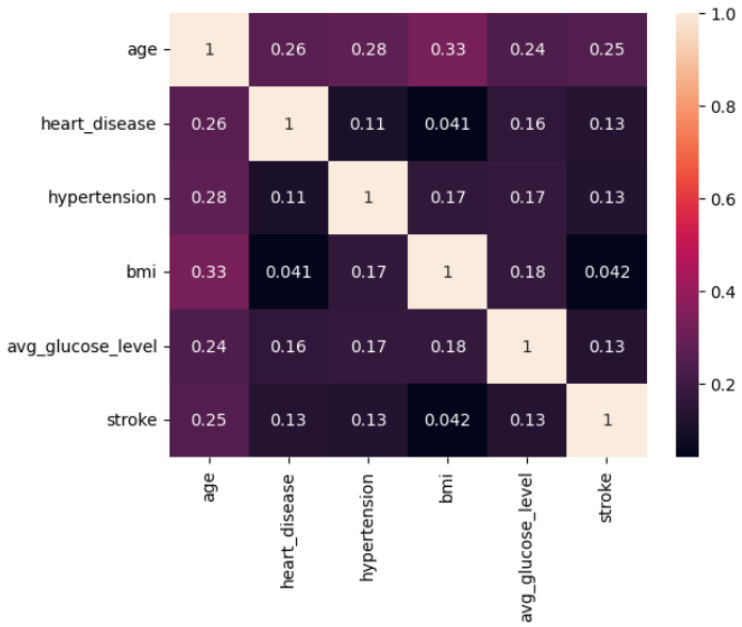
We oberve that we do not have a balance data set according to the amount of strokes registered.

We have 250 individuals with strokes against 4800 individuals without a stroke



Linear correlation

Studying the correlation between variables we have



For having a stroke, we observe in decrease correlation in the following variables:

1. **age** (0.25)
2. **heart_disease**, **glucose level** and **hypertension** all three with (0.13),
3. **bmi** (0.04)

Data exploration

During the data exploration (see *jupyter notebook* for more details) we observe that:

- The data regarding **smokers** seems to not have a direct impact on the chances of having a stroke
    - For this reason we have erased this variables.
- The **bmi** is not complete and if we take the mean of the NA values we could bias the data.
    - For this reason we have dropped the NA values.
- Not a relevant difference between **male** and **females** affecting a stroke
- **Hypertension, age** and **heart diseases** have a direct relationship with the chances of having a stroke
- When grouping into clusters, the **glucose levels** above the D07 group (diabetes), have a higher chance of having a stroke.
- Regarding the job type, being **self-employed** seems to have a higher impact on the chances of stroke

Feature selection

The studied feature selection strategies are (see notebook):

- Selecting from the correlation matrix
- Mutual Information
- Logistic Regression with a forward step strategy
- Random Forest with a forward step strategy
- Variable importance through Random Forest model

After studying these feature selection strategies, we decided three feature selection strategies on several models (see *Model selection* part):

1. ***Forward step* (SFS) using a linear *regression model***
2. **Variable importance** with ***RandomForest with a 1-SE criteria***
3. **A *manual selection*.** This manual selection was done according to the linear correlation and the analytical study. The manually selected variables were:

    - Age
    - Heart disease
    - BMI
    - Hypertension
    - Average Glucose Level
    - Work type: self-employed

After comparing these 3 strategies with also each model, we arrive to the conclusion that:

We observe that **Random Forest Variable Importance + 1SE** is the best feature selection for:

1. Random Forest
2. MLP Classifier

We observe that **Manual feature selection** (*age, heart_disease, bmi, hypertension, avg_glucose_level, work_type_Self-employed*) is the best feature selection for:

1. Logistic Regression
2. SVC

Model selection

Once we have done the best feature selection, now, we are going to compare the following **models** for *classification*:

1. Logistic Regression
2. RandomForest
3. SVC
4. MLPclassifier

At the same of selecting the model, we are going to select the best feature strategy:

1. Select a model
2. **Train** the model against the **validation set**

3. Select the best scored strategy in validation set between the proposed feature selection strategies.

For the **parameter tuning** we are going to use **GridSearchCV**. For the purpose of *tuning the models*, we are going to use a **pre-defined Split** that it is the union of:

1. The SMOTE data as a **training set**. This set is used for training the models.
2. The **Validation Set** that we had separated previously. We are going to tune the parameters and select the ones associated to the best score on this set.

Finally, we are going to evaluate the results with respect to our **testing data set**.

Model evaluation

For the evaluation we have splited the data into *training*, *validation* and *test*.

We are going to use:

1. the *training* set for *SMOTE* **after the transformation**
2. a *validation* set for *fine-tuning*
3. and *test* set for the final step

Notice that, we want to validate without any synthetic data in the *test* and *validation* set.

We have focused the SMOTE on the training set because we have observed that if we smote all the data set, the results will tend to overfit the data (see §2.7.0 in jupyter notebook).

When dividing the data set in three, we are going to use the following ratios:

- Training set: 70%
- Validation set: 15%
- Test set: 15%

Model tuning

For the model tuning we are going to use the **validation set**

Model validation

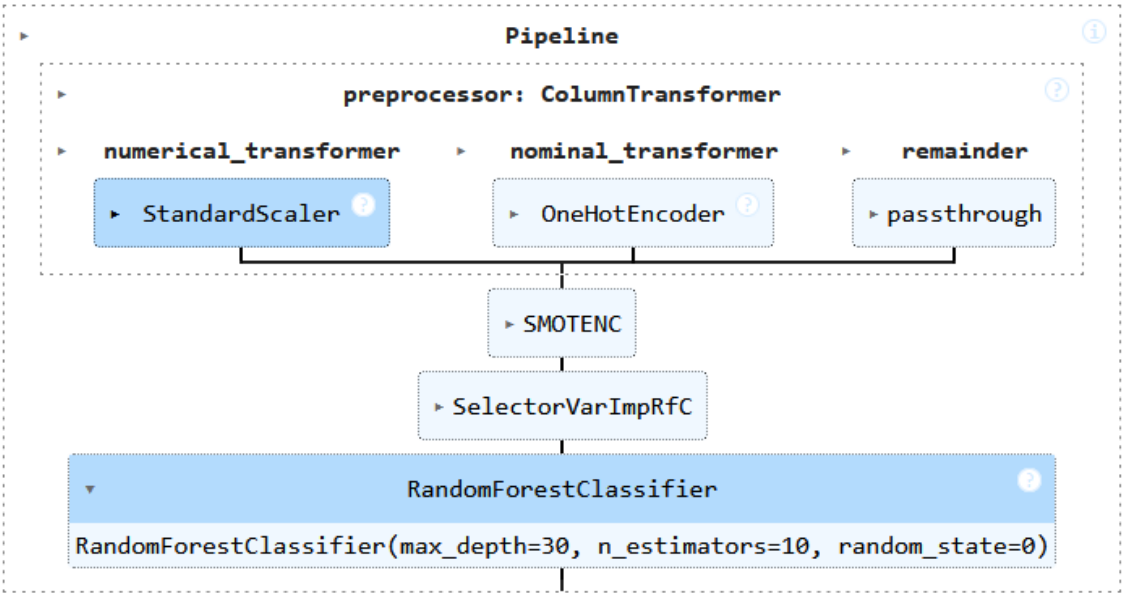
For the model evaluation we are going to use the **test set**

Best Model

The best model was a **Random Forest with an accuracy of 85%** and with the following parameters:

- Max_depth = 30
- Min_samples_leaf = 1
- N_estimators = 10

NB: The process pipeline throws us a best estimator that corresponds to the same model and the same model parameters:



Git repository

The URL for the github repository is:

https://github.com/zBotta/strokesML