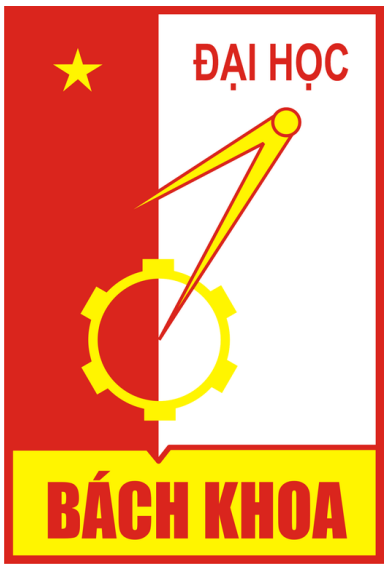
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY



REPORT

MACHINE LEARNING AND DATA MINING

IT3191E - 139399

STROKE PREDICTION

Instructor: TS. Nguyen Nhat Quang

Group: 13

Nguyen Kieu Trang - 20205174

Mai Thi Ngoc Anh - 20205143

Hanoi, June 2023

Contents

1 Definition

1.1 Project overview

Stroke is one of the major causes of death. According to the World Health Organization (WHO), stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.

In this project, we're building a model capable of early predicting whether a patient is likely to get a stroke or not. The prediction is made by learning from thousands of patients. Each patient's information includes gender, age, smoking status, hypertension status, marital status, etc.

Our goal is to get familiar with doing experiments in DS and ML, understand the workflow of a project, and get insight from the dataset. We select CART algorithms for this project. We don't have enough time to build everything from scratch so we'll take advantage of various tools from the Scikit-learn, Pandas, Numpy, and Imbalanced-learn libraries.

1.2 Problem Statement

The tasks involved are the following:

1. Download the Stroke dataset from Kaggle: `https://www.kaggle.com/fedesoriano/stroke-prediction-dataset`
2. Do basic data preparation including data cleaning.
3. Test the impact of different data transforming and sampling techniques on each model's performance.
4. Tune each model's parameters.

1.3 Test Option and Evaluation Metric

We'll use Repeated Stratified 5-fold Cross Validation to estimate F1 score.

$$F_1 = 2\frac{Precision*Recall}{Precision+Recall}$$

Where $Precision = \frac{TruePositive}{TruePositive+FalsePositive}$ and $Recall = \frac{TruePositive}{TruePositive+FalseNegative}$

This metric is selected because our dataset is severely imbalanced (see Pigture1)

Among 5110 records, 0 accounts for 4861 records (95.1%), and 1 accounts for 249 records (4.9%). This shows that our data is severely imbalanced.

We use a validation set extracted from 10-time Repeated Stratified 5-fold Cross-Validation.

Avoid misleading evaluation results: A 5-fold cross-validation is appropriate for an imbalanced dataset because a fold is ensured to be a representative sample of the domain.

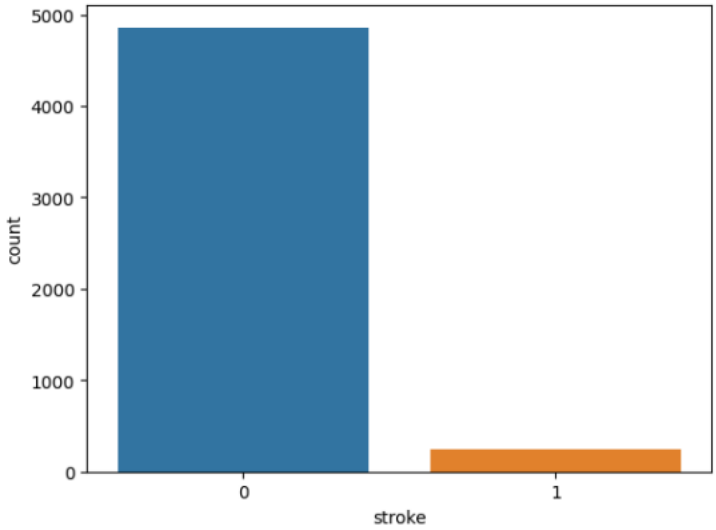


Figure 1: 'Stroke' distribution

2 Data Analysis

2.1 Data Exploration

The Stroke dataset has 5110 records, each record has the following fields:

- id: a unique identifier (int)
- gender: "Male", "Female" or "Other" (string)
- age: age of the patient (float). Min: 0.08, Max: 82.0
- hypertension: 0 for not having hypertension, 1 for having hypertension (int)
- heart_disease: 0 for having heart disease, 1 for having heart disease (int)
- ever_married: "No" or "Yes" (string)
- work_type: "children", "Govt_jov", "Never_worked", "Private" or "Self-employed" (string)
- Residence_type: "Rural" or "Urban" (string)
- avg_glucose_level: average glucose level in the blood (float). Min 55.12, Max: 271.74
- bmi: body mass index (float). Min: 10.3, Max: 97.6
- smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown" (string)

- stroke: 1 if the patient had a stroke or 0 if not (int)

Each of these attributes is observed in 5110 records, except for ‘bmi‘ which has 4909 records observed. This implies that ‘bmi‘ is having a fair number of missing values.
By observing the unique values of each attribute, we can easily split these attributes into:

- Numerical variables: ‘age’, ‘avg_glucose_level’, ‘bmi’
- Categorical variables: ‘smoking_status’, ‘gender’, ‘hypertension’, ‘heart_disease’, ‘ever_married’, ‘work_type’, ‘Residence_type’

2.2 Exploratory Visualization

Down here we show some figures worth mentioning about numerical variables.

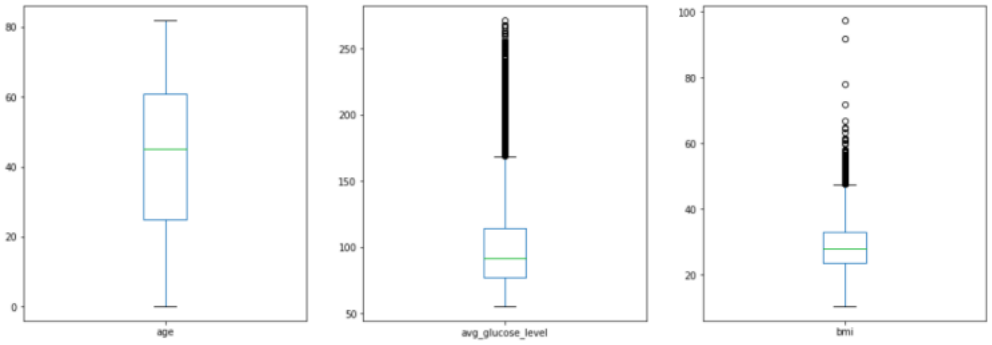


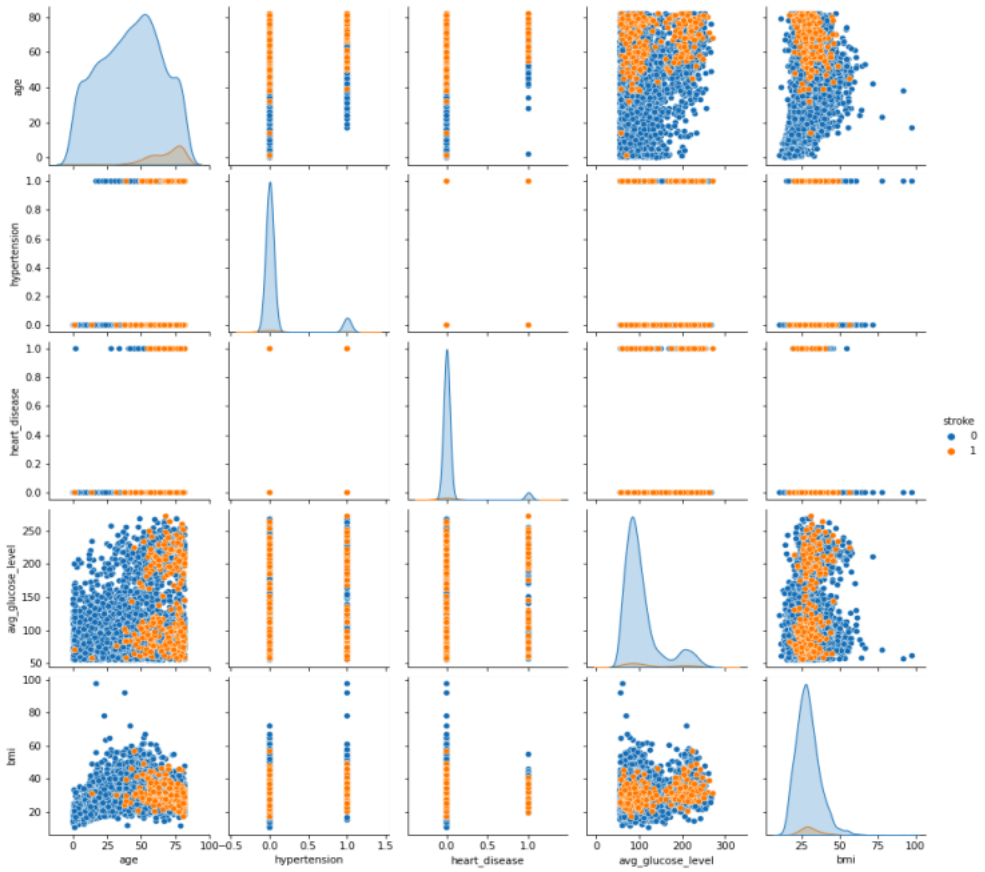
Figure 2: Box and whisker plots for ‘age’, ‘avg_glucose_level’, ‘bmi’

	age	avg_glucose_level	bmi
count	5109.000	5109.000	4908.000
mean	43.230	106.140	28.895
std	22.614	45.285	7.854
min	0.080	55.120	10.300
25%	25.000	77.240	23.500
50%	45.000	91.880	28.100
75%	61.000	114.090	33.100
max	82.000	271.740	97.600

We pay attention to the ‘bmi’ whose several records are quite far from others. This suggests they could be outliers and possibly need removal (See the Data Preparation subsection).Fig. 3: Scatter pair plot with respect to the ‘stroke’ attribute (“pairplot.png”). **Note**: all values 0 of stroke are put BEHIND values 1 before plotting, indeed they OVERLAP each other.

By eye, we can't find any single attribute that can clearly classify 'stroke'. The only characteristic we can realize that most stroke patients whose 'age' is greater than 50 and whose 'bmi' is smaller than 50.

2.3 Algorithms and Techniques



3 Methodology

3.1 Data Preparation

In this Machine Learning course, we're not going to spend much energy in data preprocessing, because it seems more relevant to the Data Science course. Instead, we'll mostly focus on model-centric.
The preparation steps are done in cart.ipynb, which includes:

1. **Remove label noise and outliers (using Quantile Range Method).**
2. **Split the dataset into a training set and test set (using Stratified train_test_split).**
3. **Do data transformation.**
    - Discretize numerical variables:
        - As far as we know, age and bmi can be separated into 4 groups each:
            - age:
                - Under 14: Children
                - 15-24: Youth
                - 25-64: Adults
                - 65 and over: Senior
            - bmi:
                - Below 18.5: Underweight
                - 18.5 – 24.9: Norma
                - 25.0 – 29.9: Overweight
                - 30.0 and Above: Obese
        - In addition, blood sugar levels can also be split into 4 group, though normal sugar level depends on each specific age group:
            - Fasting
            - Before meal
            - 1-2 hours after eating
            - Bedtime
    - Encode categorical variables:
4. **Data balance.**

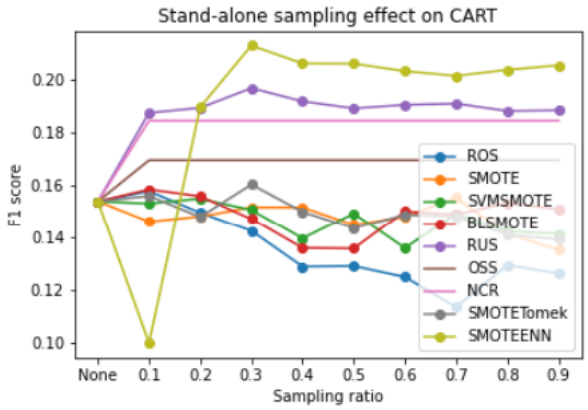


Figure 3: Effect of sampling on CART. The model used: Random oversampling, SMOTE, SVM SMOTE, Borderline SMOTE, Random undersampling, One-sided selection, Neighbourhood cleaning rule, SMOTE Tomek, SMOTE Edited nearest neighbor. #stroke: #not_stroke is set from 0.1 to 0.9 for all sampling models, except OSS and NCR.

- As we have seen, our data set is severely unbalanced. Among 5110 records, 0 accounts for 4861 records (95.1%) and 1 accounts for 249 records (4.9%).
- So the question is how do we handle it? Here we will choose to duplicate the data, here is to increase the number of 1 values. Since the attributes in the leaf class will decide whether the result is 0 or 1; and it is affected by the number of instances from each class in that leaf so if the value of 0 is too much it will affect the result.
- First we'll fix the transform method by filling in the numeric data with blank data; The transform method for categorical variables is ordinal. Selection of Decision tree model with fixed parameters.

  By different cloning methods:

  – Standalone sampling: Includes Random Oversampling, SMOTE, SVM SMOTE, Borderline SMOTE, Random Undersampling, One-sided Selection, and Neighborhood Cleaning Rule.

  – Combination of sampling: Includes SMOTE Tomek and SMOTEENN.

  After many runs, SMOTEENN shows the most promising scenario. This is unsurprising because besides balancing data via SMOTE, the technique also pays attention to the unambiguity of examples in the data set and increases the certainty of decision boundaries.

5. **Compare 2 transform methods for categorical variables.**

   For the classification tree model, there will be over-fitting.

- The reason for comparing two ordinal and nominal methods is because the two attributes have their own disadvantages:
  - ordinal: increases the number of attributes, it is possible that the generated attributes that are less important are redundant and unnecessary
  - onehot-nominal: gives rise to other relationships that can be misunderstood
- To compare these two methods, we will keep the clone method SMOTEENN and choose the Decision tree model with fixed parameters.

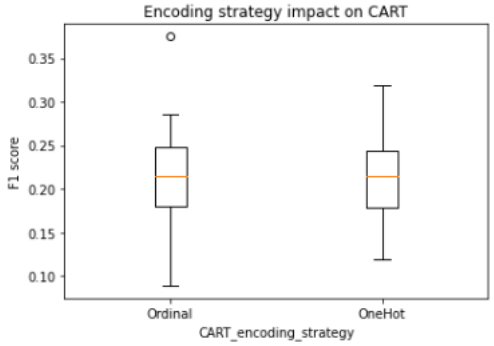


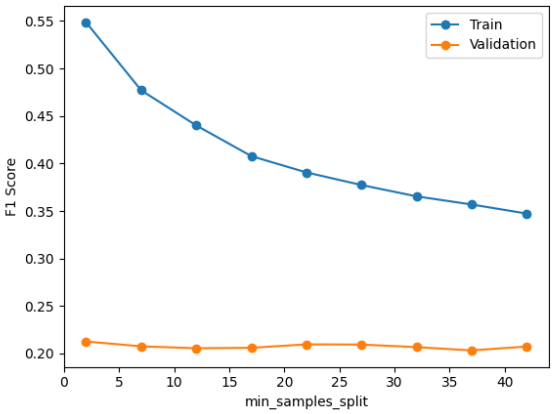
Figure 4: Encoding strategies' impact on CART. The average F1 score for Ordinal is 0.212643, and for Onehot is 0.214212.

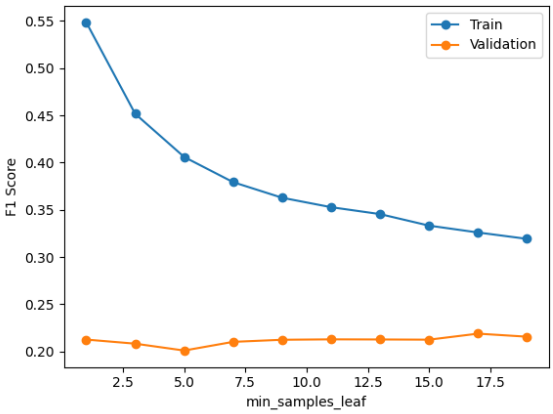
In general, with categorical variables, onehot encoding is seemingly more preferred as it does not create additional relationships. However, with decision trees, some articles claim that onehot encoding degrades the performance as it creates many more variables with less feature importance.

6. **Select model parameters - Algorithm Tuning.**

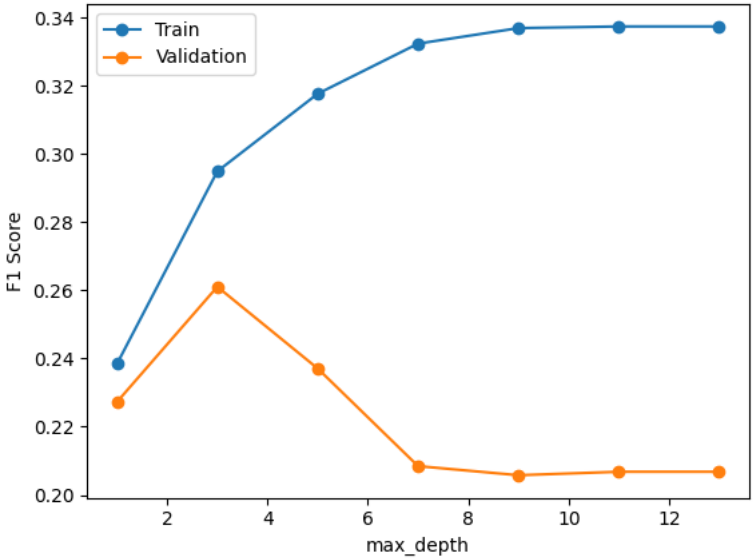
   First, we'll fix the transform method by filling in the numeric data with blank data; The transform method for categorical variables is ordinal, choosing a Decision tree model with fixed parameters and SMOTEENN with a ratio 0.3.

   We've chosen 'min_samples_split', 'min_samples_leaf', 'max_depth', 'max_features', 'ccp_alpha'.

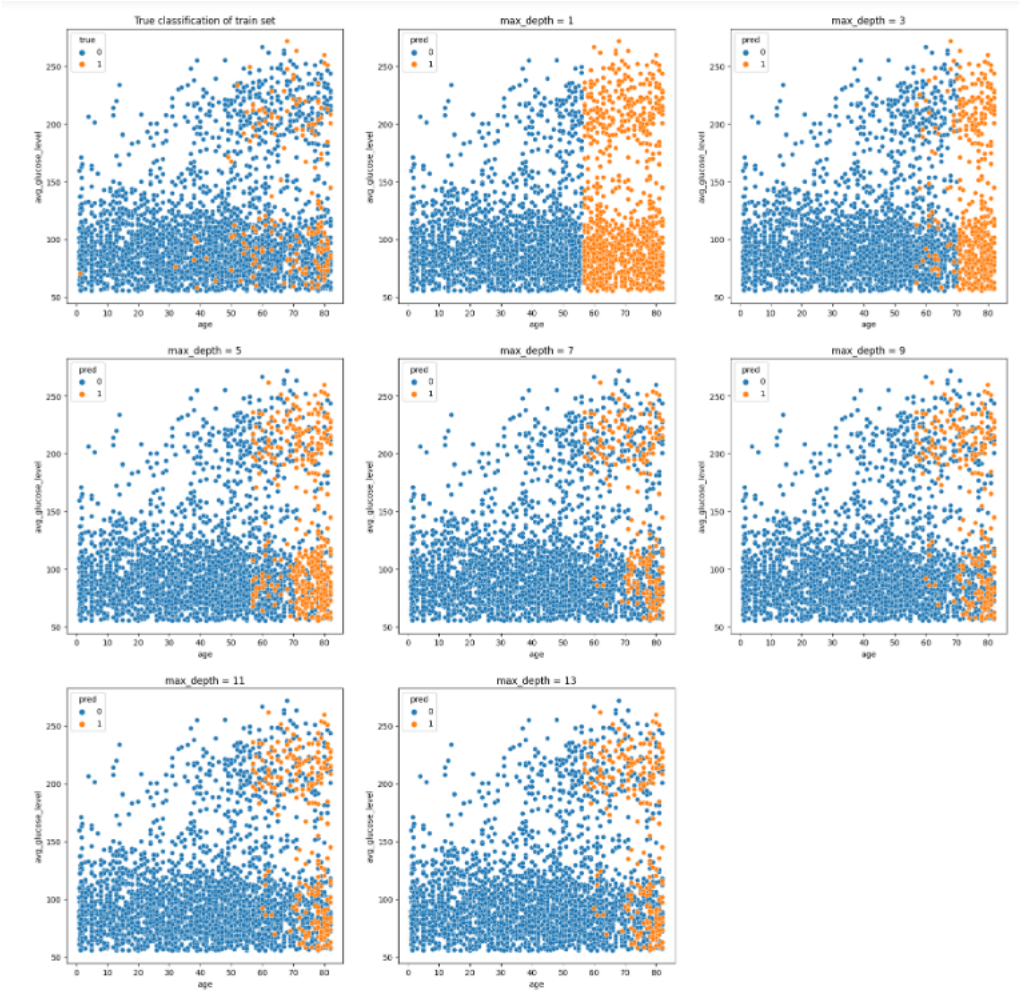




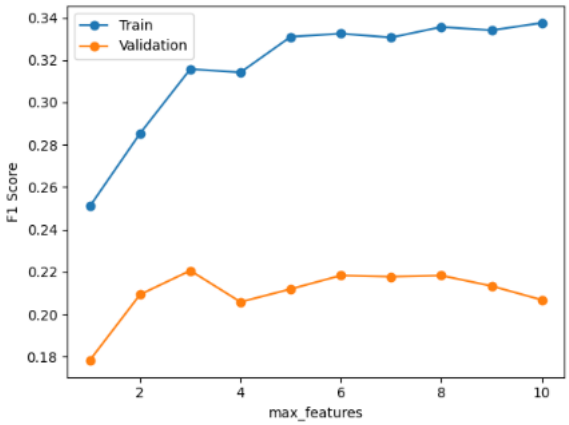
- In two parameters 'min_samples_split', and 'min_samples_leaf' we found the best 2 values in 'min_samples_split' = 37; 'min_samples_leaf' = 11.



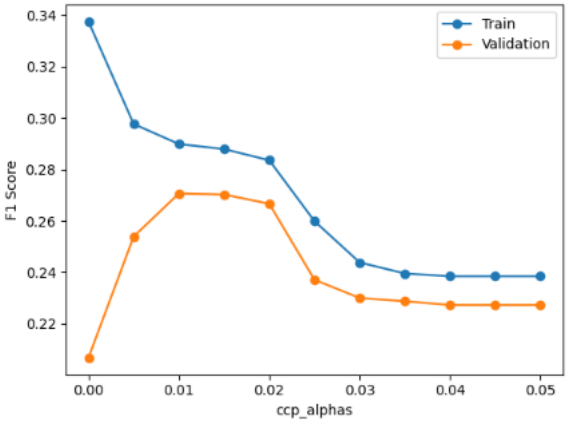
- At first glance, it can be seen that F1 increases significantly when max-depth = 5. However, in the chart below, it can be seen that when max-depth is smaller than 5, the results are not realistic.



- Max-feature and max-depth are two pre-pruning strategies to solve the over-fitting problem, so we mentioned them here.



- ccp_alpha: is a post-pruning strategy that usually works better in practice and is preferred by all.



- Here we want to select 'ccp_alpha' in the range (0.01, 0.02) because they make F1 increase significantly. However, when I plot the data on each value as above 'ccp_alpha' > 0.006 the predictions made will be unrealistic and of lower quality .

From the above parameter predictions, we get 'min_samples_split' = 37; 'min_samples_leaf' = 11, and 'ccp_alpha' = 0.001. Here we do not specify 'max-depth' and 'max-feature' anymore because when examining these two parameters, they do not give satisfactory results. And instead of specifying 2 pre-pruning parameters, we choose to specify the 'ccp_alpha' parameter as post-pruning which will show better results.

4 Results

4.1 Model Evaluation

With CART, on the final test set, the classification report is described as below:

```
CART training time: 0.0670318603515625s
CART predicting time: 0.006533145904541016s
[[1487  102]
 [  57   25]]
              precision    recall  f1-score   support

           0       0.96      0.94      0.95      1589
           1       0.20      0.30      0.24        82

    accuracy                           0.90      1671
   macro avg       0.58      0.62      0.59      1671
weighted avg       0.93      0.90      0.91      1671
```

4.2 Explanation

Although we exploit a variety of techniques to deal with imbalanced data and configure the model, our overall F1 results do not exceed 0.25, which is quite low. One explanation for this situation could be the fact that the stroke dataset is very overlapping and unbalanced.

4.3 Conclusion

The model gives a rather low f1 score of 0.24, but it has a fairly fast prediction time. The shopping cart is also one of the models that is likely to improve clearly in the future. And with many data processing techniques offered, the results are also not well improved, so this dataset is not reliable and accurate enough for medical problems.
Look at the bright side, even when the models are not ready to use, we have obtained some good experiences during project time.

- Firstly, when dealing with seriously imbalanced data, we have to search for the solution to this problem and find the Imbalance Handling technique. And clearly this technique has significantly improved all of our models. This experience in handling unexpected situations is absolutely useful for us in the future.

- One other benefit we received during the project is the skill of visualizing data and experimental results as well as the ability to do researches on other articles, libraries and documents.

- We also have to draw comments on our dataset, our practical problem to make decisions during the project (Using f1 score for measurements instead of accuracy, focus on recall rather than precision rate).

- We also have to draw comments on our dataset, our practical problem to make decisions during the project (Using f1 score for measurements instead of accuracy, focus on recall rather than precision rate).

Along with some useful skills and experiences, we still struggle with some hardships:

- Most of us haven't got used to the workflow and team working on building a model. The difficulties in handling unexpected problems make us confused for a while, also this is our first project so we aren't sure if the results are good enough, or if the data is handled correctly.

- Another thing is that our report and experimental results are not in a good technical writing and can lead to misunderstanding for readers.

5 References

https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifie

html

https://imbalanced-learn.org/stable/under_sampling.html

https://imbalanced-learn.org/stable/over_sampling.html

https://imbalanced-learn.org/stable/combine.html

https://towardsdatascience.com/one-hot-encoding-is-making-your-tree-based-ensembles-w