Prediction of Stroke Disease

Report of ML01 Project

By

Jing Zukuan and Wang Hongzhe

18124527 and 18124533

2021/5/30

Introduction

We find a dataset whose each row in the data provides relevant information about the patient who has a stoke.

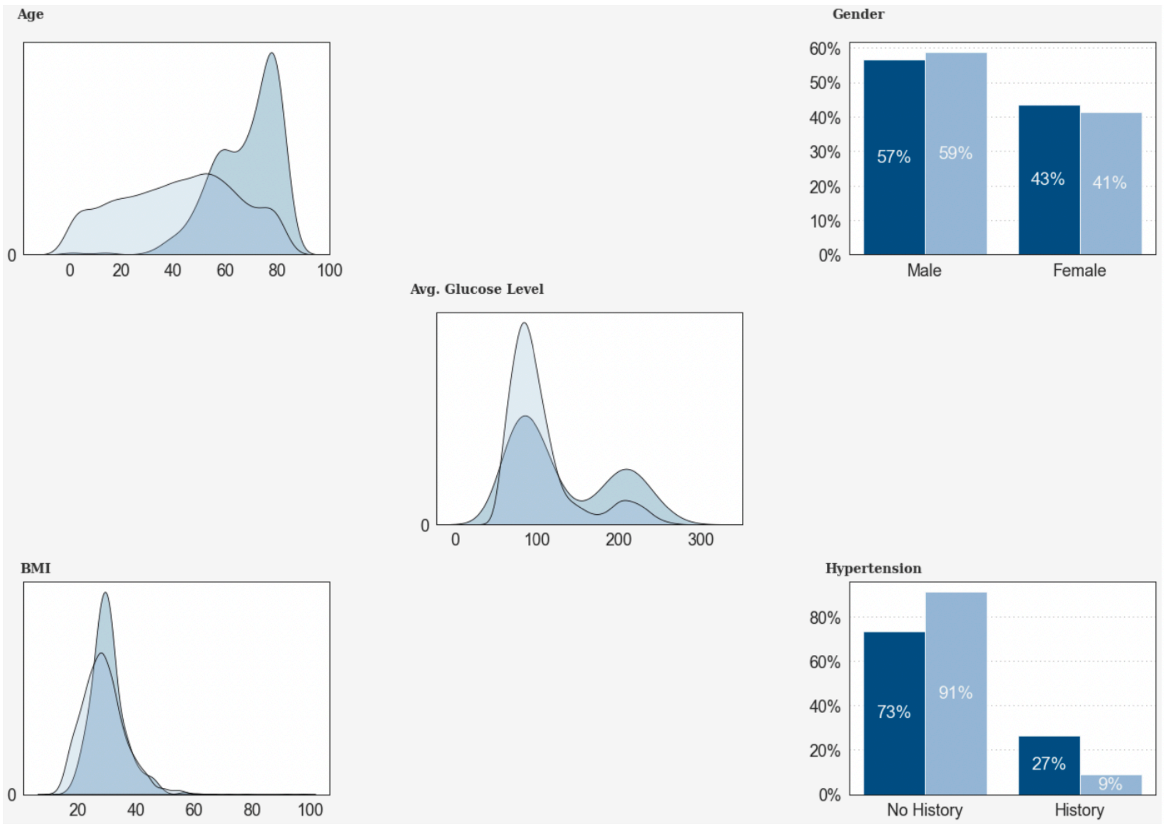
There are 12 attributes in the dataset, we use the dataset to predict whether a person is likely to get stroke.

The dataset can be find among following URL:

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

Step 1: Handle the data

After having a simple glance at the dataset, there exists some null data of the parameter BMI. Since the scale of the dataset is too small, it isn't reasonable to delete the null data directly. By the way, I predict these empty data by 'age' and 'gender', using decision tree.

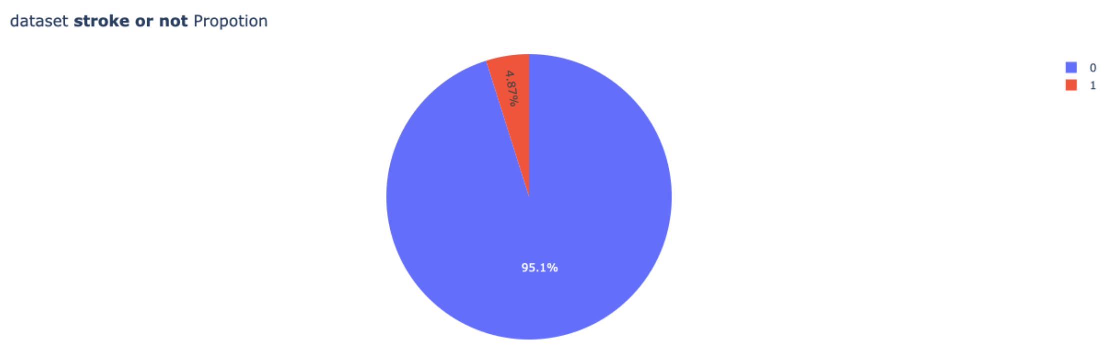


Step 2: Find the most relative feature

I plot the distribution of every feature and 'stroke', and I draw the heat map among every input. I find the occurrence of stroke disease is highly related to 'Age', 'Gender', 'BMI', 'Hypertension' and 'Average Glucose Level'. So I choose these attributes to train the model.

Step 3: Balance the proportion of positive and negative cases

We found that in this dataset, 95% of the cases are negative, which means that peoples who donate these datas never got stroke.



The huge distinction of the proportion will make a high rate of inaccuracy.

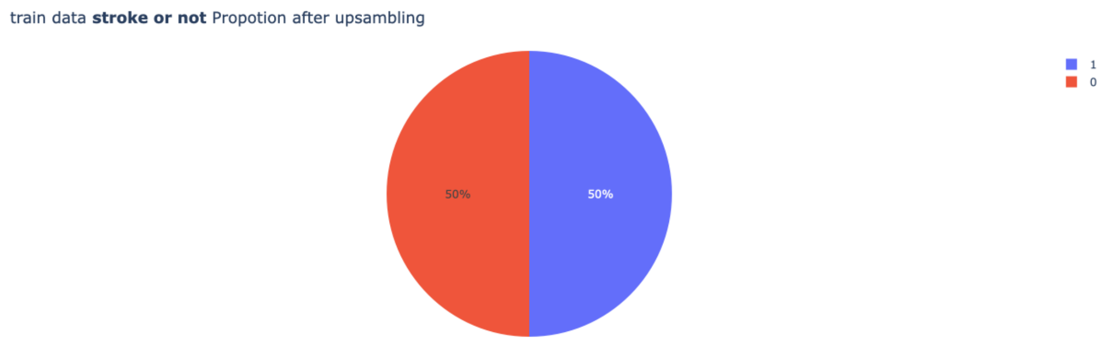
Owing to the small amount of the dataset, I use over-sampling to balance the proportion.

However, the traditional over-sampling method, just duplicating these minority sample and making no new information to the dataset, which will easily causes over-fitting problem. In this case, I chose SMOTE (Synthetic Minority Over-sampling Technique) to generate new datas.

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line. As the function following:

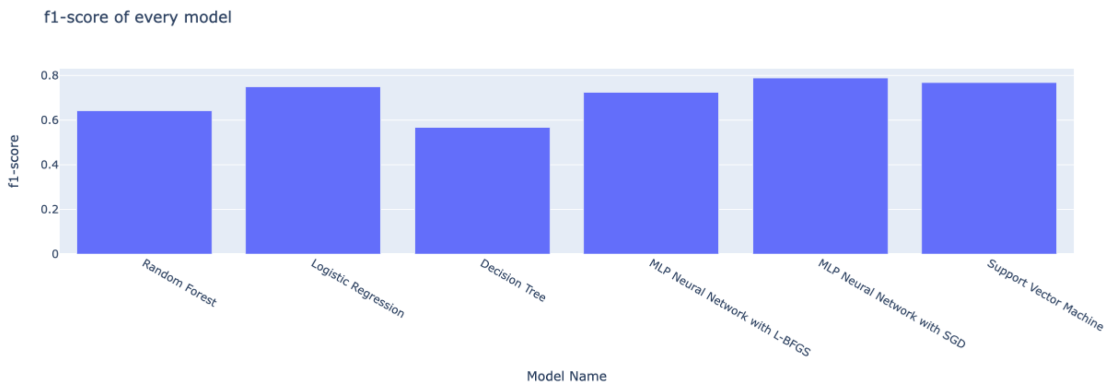
$$x_{new} = x + rand\,(0,1) \times (\hat{x} - x)$$

And then we have a dataset having 50% of positive sample and others are negative.



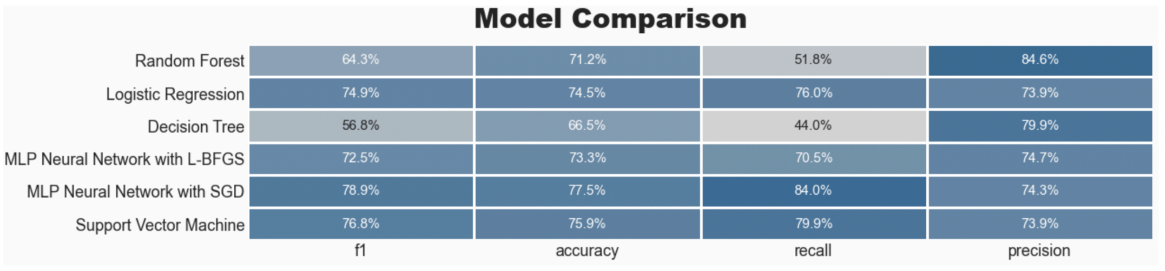
Step 4: Model Selection

For comparing, I select 6 models, using cross validation to get the accuracy of every model. We found that 'Random Forest' leads far ahead than other models.



However, sick people don’t want to get a healthy answer when predicting, so we should more focus on ‘Recall Rate’ of predicting stroke.

By the way, I found that ‘MultiLayer Perception Network with SGD algorithm’ performs best on recall rate.



Model Comparison

	f1	accuracy	recall	precision
Random Forest	64.3%	71.2%	51.8%	84.6%
Logistic Regression	74.9%	74.5%	76.0%	73.9%
Decision Tree	56.8%	66.5%	44.0%	79.9%
MLP Neural Network with L-BFGS	72.5%	73.3%	70.5%	74.7%
MLP Neural Network with SGD	78.9%	77.5%	84.0%	74.3%
Support Vector Machine	76.8%	75.9%	79.9%	73.9%

Step 5: Optimize the parameter

Because of choosing MLP, we can do early stopping when we get the highest negative recall rate. And we can also reset the hidden layers and output cells.

Finally, I chose the network of one hidden layer with 5 neural units and an output layer of 2 neurons. The iterate time is set to 23.

Conclusion

Due to the randomness of the method stochastic gradient descend and we can do early stopping when we training the data with MultiLayer Network, we can control when to stop by the will of my own.

So in this case, I can gain a higher recall rate by sacrificing the accuracy rate.

But if we want the best performance of accuracy, (balanced) random forest is still the best choice.

Others

The code (notebook) can be found on GitHub:

https://github.com/Raymond-47/Project_ML01.git