**Report Machine Learning**

Decision tree & Random Forest

# Topic: Stroke Prediction

## Context

* According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.
* The selected dataset is used to predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status. Each row in the data provides relevant information about the patient.

## Attribute information in the dataset

1. id: unique identifier
2. gender: "Male", "Female" or "Other"
3. age: age of the patient
4. hypertension: 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
5. heart\_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
6. ever\_married: "No" or "Yes"
7. work\_type: "children", "Govt\_jov", "Never\_worked", "Private" or "Self-employed"
8. Residence\_type: "Rural" or "Urban"
9. avg\_glucose\_level: average glucose level in blood
10. bmi: body mass index
11. smoking\_status: "formerly smoked", "never smoked", "smokes" or "Unknown"\*
12. stroke: 1 if the patient had a stroke or 0 if not

# Theory

## Model

1. Decision tree
2. Random forest

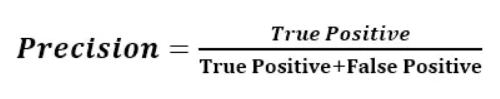
## Evaluate

1. Confusion matrix: [[TN, FP], [FN, TP]]
2. Model testing accuracy

* This is the simplest math problem, calculated by dividing the number of correct predictions by all predictions.
* The disadvantage of this evaluation method only tells us what percentage of data is classified correctly without showing specifically how each type is classified, which class is classified correctly the most or the data. of which class is most often misclassified into other classes K-Fold Validation Mean Accuracy

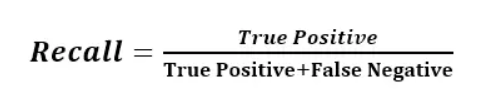
1. Precision score

* Precision will tell us how many Positive predictions are actually True.



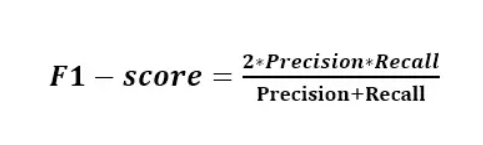
1. Recall score

Recall is also an important metric. it measures the rate of correctly predicting positive cases across all samples in the positive group.



1. F1 score

* Depending on the problem, you will want to prioritize using Recall or Precision. But there are also many problems where both Precision and Recall are important. A popular metric that combines both Recall and Precision is called F1-score.

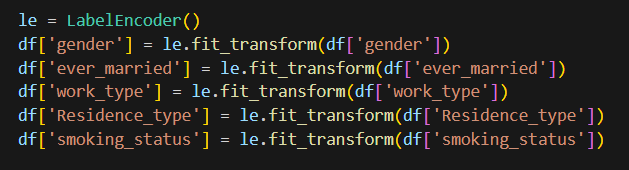


# Decision Tree Model

## Library

## Preprocessing

* 1. Label encoding is a technique use to convert categorical labels into numeric form so that they can be provided to machine learning algorithm to improve predictions.



## Training & improvement

### Initially #0

* Score
  + Confusion Matrix: [[927, 45], [ 44, 6]]
  + Model testing accuracy: 91.29158512720157 %
  + K-Fold Validation Mean Accuracy: 90.56%
  + Precision Score: 11.76%
  + Recall Score: 12.00%
  + F1 Score: 11.88%
* Comment
  + Although the model testing accuracy and K-Fold Validation Mean Accuracy is high, precision score, recall score and F1 score is much lower. This mean that the model can predict TN well however it is bad for predict TP.
  + The cause of this problem lies in the fact that in the dataset, the number of negative values is much greater than the number of positive values.

### Improvement #1: oversampling with smote()

* For addressing problem in #0state, I make use of SMOKE from imblearn.over.sampling to balance dataset
  + Oversampling can be defined as adding more copies to the minority class.
  + SMOTE (Synthetic Minority Over-Sampling Technique) is a popular method used to address class imbalance in datasets. It works by generating synthetic samples for the minority class, thereby balancing the dataset and improving the performance of machine learning model



* Score:
  + Confusion Matrix:

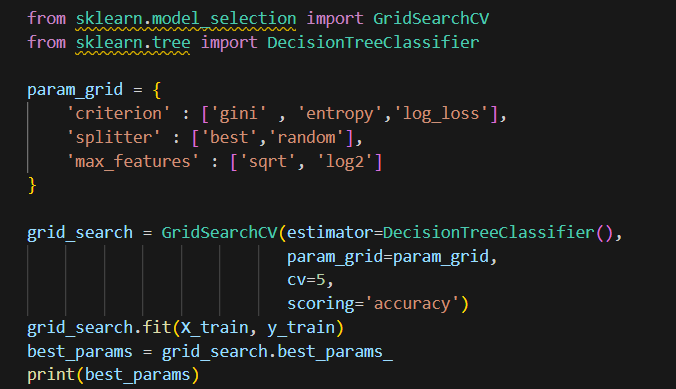
[[902, 103],

[ 64, 876]]

* + Model testing accuracy: 91.41388174807197 %
  + K-Fold Validation Mean Accuracy: 90.21%
  + Precision Score: 89.48%
  + Recall Score: 93.19%
  + F1 Score: 91.30%
* Comment
  + The model improves dramatically after implementing oversampling in the dataset. However, model testing accuracy decrease slightly (about 1%)

### Improvement #2: modification parameter in training function

* Use GridSearch to find the best parameters for model



* However, after using founded parameter, the score of the model lower than the score of default parameter
* Score
  + Confusion Matrix:

[[887, 96]

[ 82, 880]]

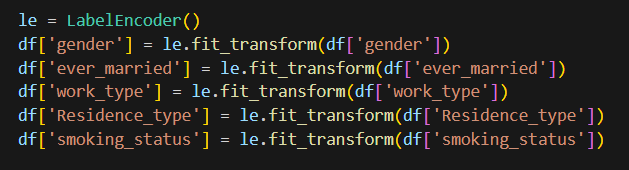
* + Model testing accuracy: 90.8483290488432 %
  + K-Fold Validation Mean Accuracy: 89.74%
  + Precision Score: 90.16%
  + Recall Score: 91.48%
  + F1 Score: 90.82%

# Random forest model

## Library

## Preprocessing

1. Label encoding is a technique use to convert categorical labels into numeric form so that they can be provided to machine learning algorithm to improve predictions.



## Training & improvement

### Initially #0

* Score
  + Confusion Matrix:

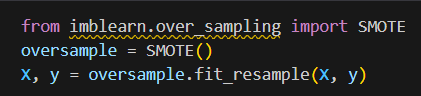
[[970, 4],

[ 48, 0]]

* + Model testing accuracy: 94.9119373776908 %
  + K-Fold Validation Mean Accuracy: 94.96%
  + Precision Score: 0.00%
  + Recall Score: 0.00%
  + F1 Score: 0.00%
* Comment
  + Although the model testing accuracy and K-Fold Validation Mean Accuracy is high, precision score, recall score and F1 score is much lower. This mean that the model can predict TN well however it is bad for predict TP.
  + The cause of this problem lies in the fact that in the dataset, the number of negative values is much greater than the number of positive values.

### Improvement #1: oversampling with smote()

* For addressing problem in #0state, I make use of SMOKE from imblearn.over.sampling to balance dataset
  + Oversampling can be defined as adding more copies to the minority class.
  + SMOTE (Synthetic Minority Over-Sampling Technique) is a popular method used to address class imbalance in datasets. It works by generating synthetic samples for the minority class, thereby balancing the dataset and improving the performance of machine learning model



* Score:
  + Confusion Matrix:

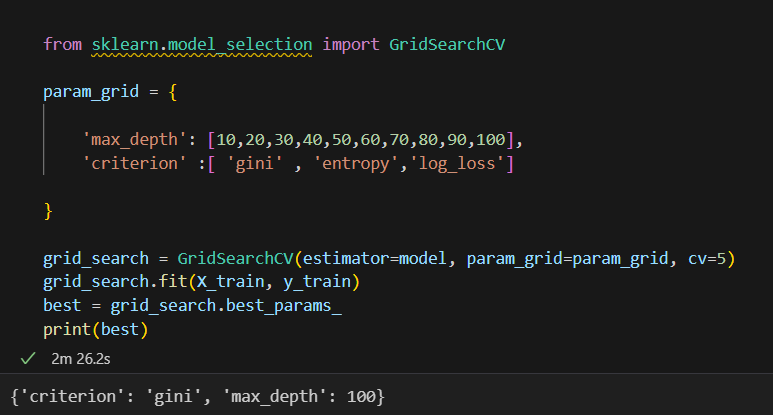
[[913, 59],

[ 37, 936]]

* + Model testing accuracy: 95.0642673521851 %
  + K-Fold Validation Mean Accuracy: 93.92%
  + Precision Score: 94.07%
  + Recall Score: 96.20%
  + F1 Score: 95.12%
* Comment
  + The model improves dramatically after implementing oversampling in the dataset.

### Improvement #2: modification parameter in training function

* User GridSearch to find the best parameters for model



* In the trying above, I have tried to find some best value for parameter max\_depth and criterion, however, the best result found is the same as the default value of the model.