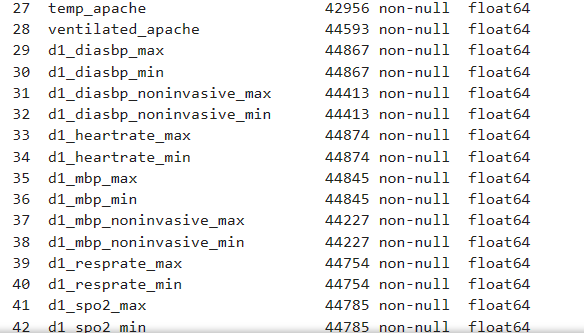
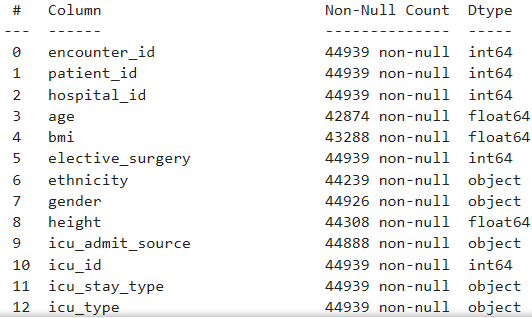
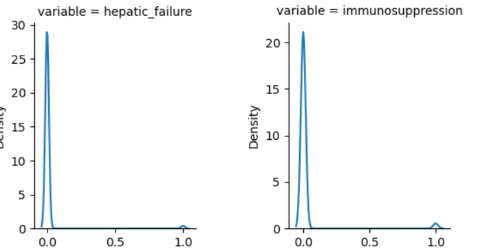
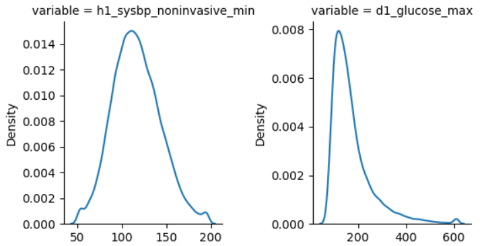
# Data mining assignment2

### Data pre-preprocessing

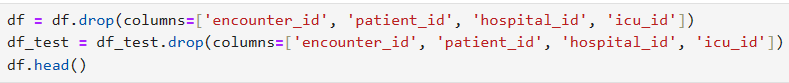
#### Dataset analysis:



First, I checked about the features of the dataset, discover that there are some types of features of majority, **categorical, continuous, and binary**. Below is frequency distribution of continuous and binary data.

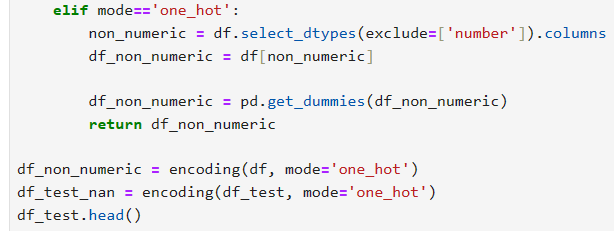


#### Data cleaning:



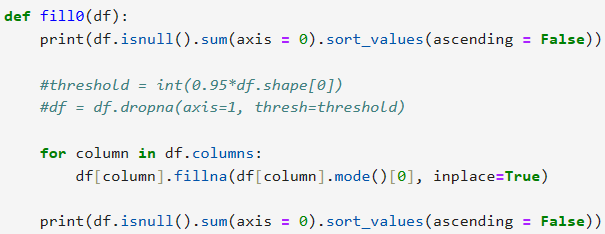
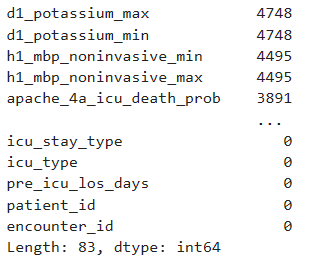
Secondly, I deleted data mentioned above, since they are just some ID, provide no information about patient’s health condition, give huge noise while doing ML process.

#### Data transformation:



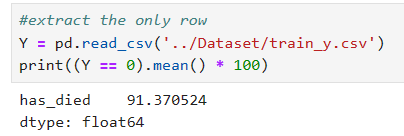
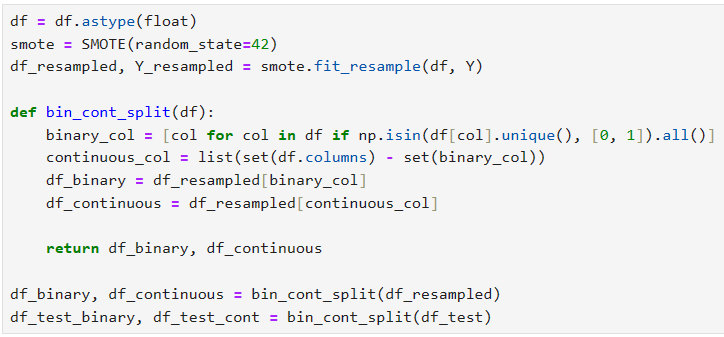
Then, I decided to do **one-hot encoding**, to **convert categorical data into binary.**

#### Data imputation:



I checked about the number of 0s in those column having missing values, there’s no column that loss data too much, so I didn’t prune any column, but just fill these NaN with **mode** of the column.

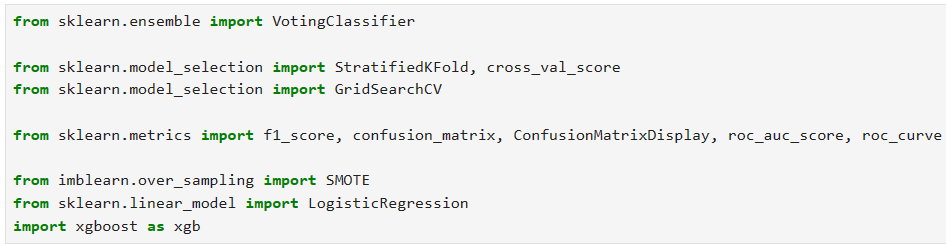
#### Data imbalance handling:



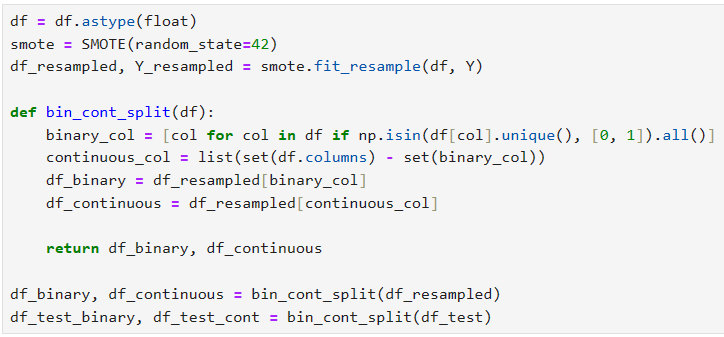
There is obvious data imbalance, for not to lose any important information from those data with result 0, my method to handle it is **oversampling**. And to prevent overfitting because of repeatedly calculate same data, I used **SMOTE** method to synthesis data.

### Classification Methods:

Below are my functions used for data classification:



Firstly, I split the resampled data into two part: **data with binary values** (including one-hot encoded categorical data) and **those with continuous values**.



Then I threw these two part into **XGBClassifier** and **LogisticRegression**, and using grid search with cross validation to find their best hyper-parameter respectively.

After, these two model is put into a **soft voting classifier**, below is the structure:

**Dataset**

**Binary**

**data**

**Xgboost**

**classifier**

**Logistic**

**regressor**

**Continuous**

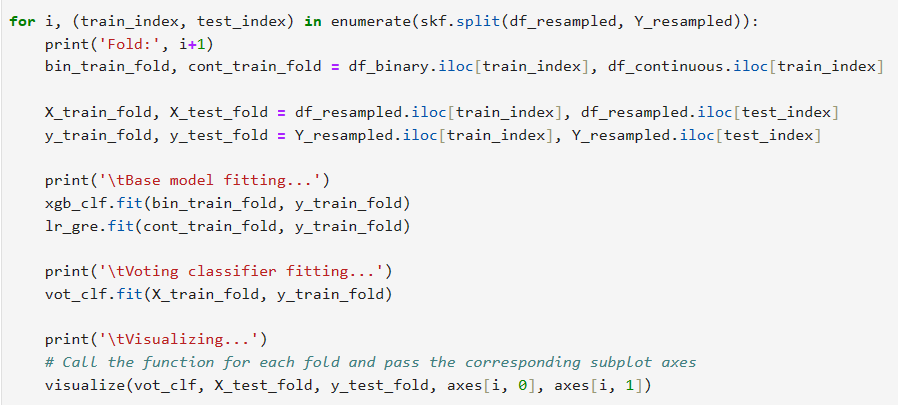
**data**

**Soft voting**

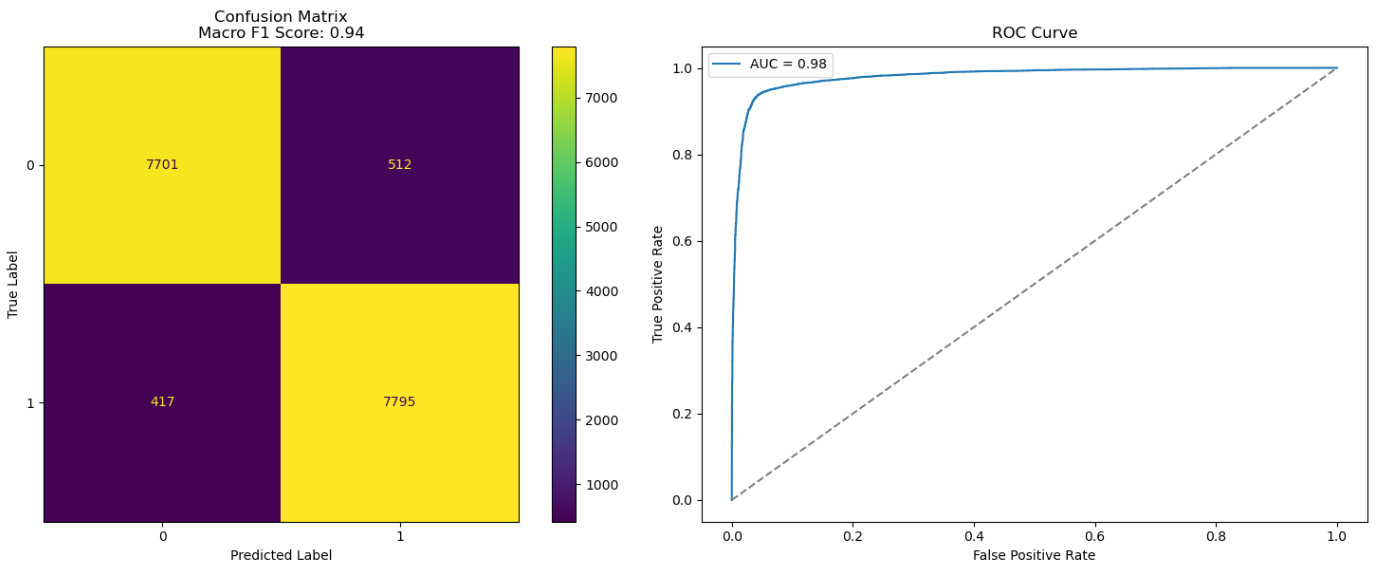
**classifier**

**Output**

Lastly, we will use cross validation, to be careful, despite over sampling, **I always use stratified k-fold to cut it into 5 slices**. Because of model complexity, **I don’t use cross\_val\_score directly**. Train two base model first, then train the voting classifier. A base model doesn’t use unseen features for prediction, and voting classifier doesn’t re-fit the base model if it is pre-fitted, which makes this kind of multi model structure achievable.



### Results & Analysis:

Below is the visualization of validation result, I use **stratified-k-fold** on **original dataset** (not oversampled), which will be shown every fold.

The high macro f1-score and AUROC is maintained in another 4 Fold.

Below is my top20 features for classification (**one-hot encoding backtracked**):

