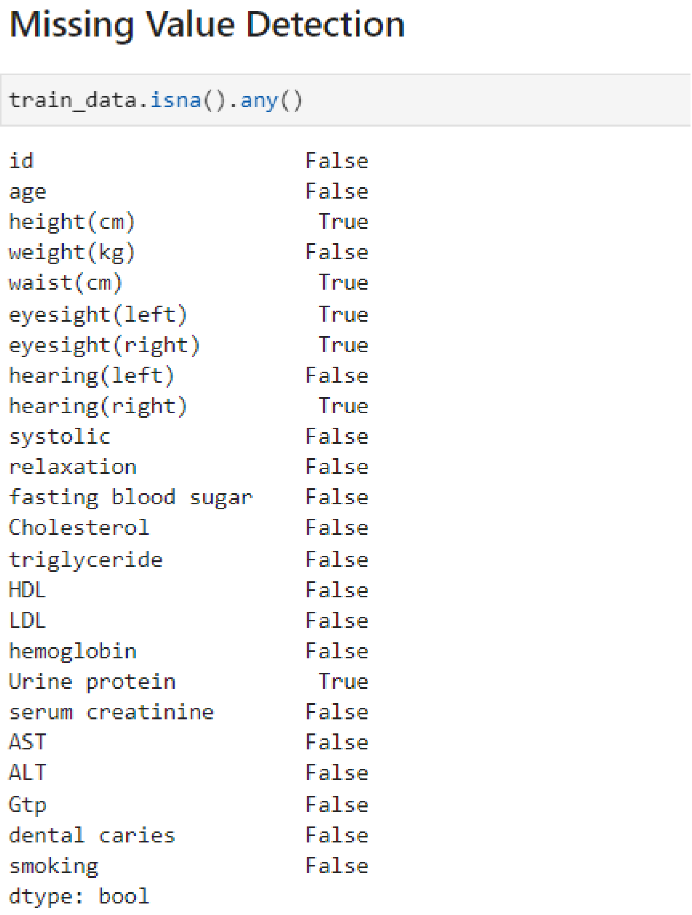
Q5-Smoke Status Recognition

This Task can be basically regarded as a binary classification problem. To have a more accurate result, Ensembling Machine Learning methods are always the first choice. Thus, I use the XGBOOST algorithm to tackle this binary classification problem.

Data Manipulation and Analysis

Firstly, I do some data manipulation and analysis to have a deep understanding of our dataset. The total training sample counts 159256, and there are some samples containing the missing value.



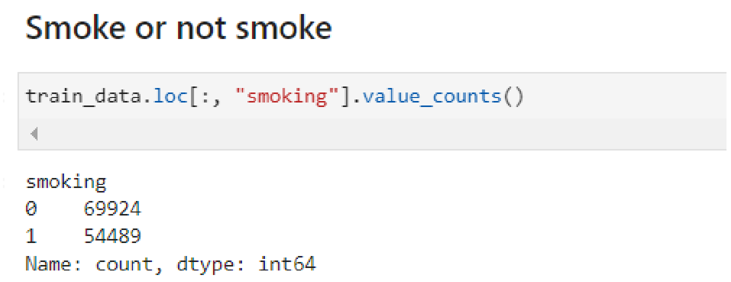
Missing Value Detection

```
train_data.isna().any()
```

```
id                     False
age                    False
height(cm)              True
weight(kg)             False
waist(cm)               True
eyesight(left)          True
eyesight(right)         True
hearing(left)          False
hearing(right)          True
systolic               False
relaxation             False
fasting blood sugar    False
Cholesterol            False
triglyceride           False
HDL                    False
LDL                    False
hemoglobin             False
Urine protein           True
serum creatinine       False
AST                    False
ALT                    False
Gtp                    False
dental caries          False
smoking                False
dtype: bool
```

Figure 1 Missing Value Detection

We can also know the label distribution on the training data



Smoke or not smoke

```
train_data.loc[:, "smoking"].value_counts()
```

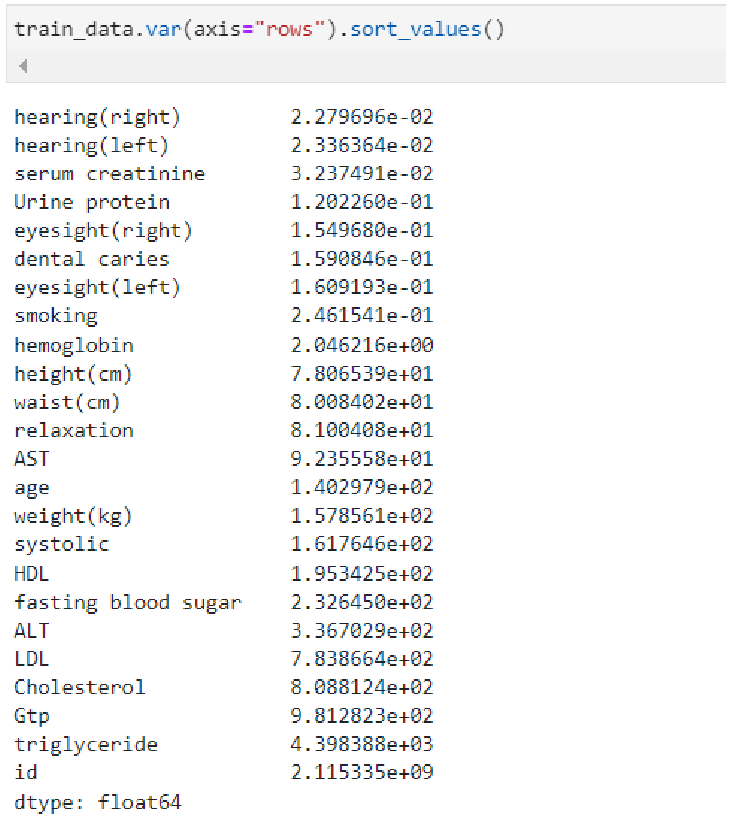
```
smoking
0    69924
1    54489
Name: count, dtype: int64
```

Figure 2 Lable Distribution

The label distribution is very close to Evenly distributed among 0 and 1

XGBoost

Before applying xgboost to classify the data, I compute the variance of each variable to see their impact on the prediction



```
train_data.var(axis="rows").sort_values()
```

```
hearing(right)         2.279696e-02
hearing(left)          2.336364e-02
serum creatinine       3.237491e-02
Urine protein          1.202260e-01
eyesight(right)        1.549680e-01
dental caries          1.590846e-01
eyesight(left)         1.609193e-01
smoking                2.461541e-01
hemoglobin             2.046216e+00
height(cm)             7.806539e+01
waist(cm)              8.008402e+01
relaxation             8.100408e+01
AST                    9.235558e+01
age                    1.402979e+02
weight(kg)             1.578561e+02
systolic               1.617646e+02
HDL                    1.953425e+02
fasting blood sugar    2.326450e+02
ALT                    3.367029e+02
LDL                    7.838664e+02
Cholesterol            8.088124e+02
Gtp                    9.812823e+02
triglyceride           4.398388e+03
id                     2.115335e+09
dtype: float64
```

Figure 3 Variance

Except id, The triglyceride variate largely among all the training sample. In addition, The ability of hearing is close for all the participants.

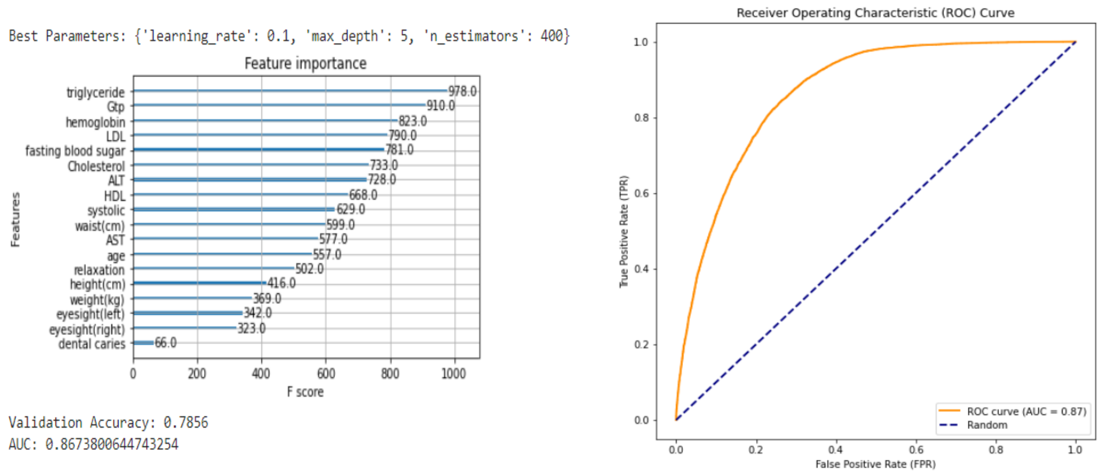
Grid Search for best parameters

I simply use the grid search method to search for the best parameters of xgboost classifier with 5-fold cross validation

- Learning rate: {0.1, 0.05, 0.01, 0.2}
- Number of estimators: {100, 200, 300, 400}
- Max Depth: {3, 5, 7}

And I also try some features selection and engineering to increase the quality of the training set.

Firstly, I try to remove the features with loweset variance.



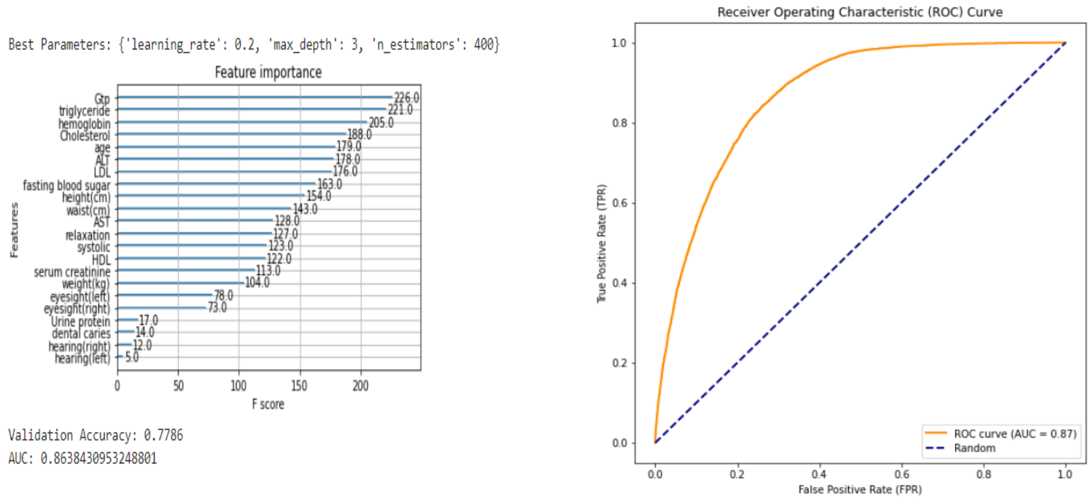
```
Best Parameters: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 400}
```

```
Validation Accuracy: 0.7856
AUC: 0.8673800644743254
```

Figure 4 Feature Importance and ROC Curve(remove features with low variance)

Average AUC on Validation dataset is 0.8674.

Secondly, I also try not to drop the missing value in training samples, which means I retain all the participants in my training sample.

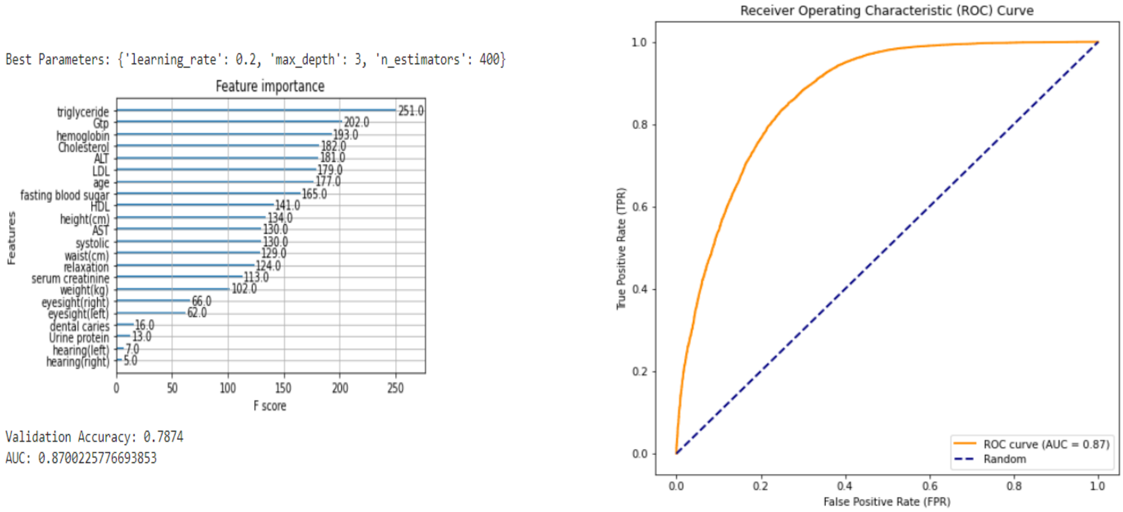


```
Best Parameters: {'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 400}
```

```
Validation Accuracy: 0.7786
AUC: 0.8638430953248801
```

Figure 5 Feature Importance and ROC Curve(without Dropna)

Lastly, I drop all the missing value in my dataset and apply the xgboost algorithm.



```
Best Parameters: {'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 400}
```

```
Validation Accuracy: 0.7874
AUC: 0.8700225776693853
```

Figure 6 Feature Importance and ROC Curve(Dropna)

This time, the AUC on validation dataset is 0.87002.

Finally, I use xgboost with the best paramters to infer on test dataset and save the results.