Capstone Project 3

Cardiovascular Risk Prediction

Satyajit Sahoo

Understanding Problem Statement:

- We are given with a data set where one of the variables represent whether a person may suffer from heart disease in next 10 years.
  Of size(3390,16)

- Our aim is to predict the same.

- The target variable is TenYearsCHD and it is binary where:

  0 means **NO**
  1 means **YES**

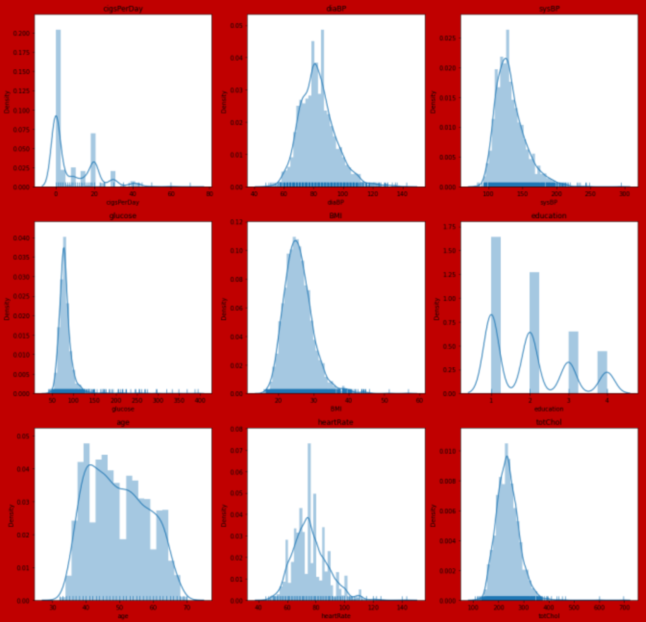
Variables Involved :

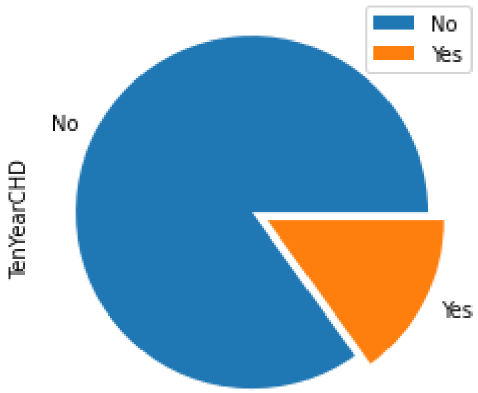
- ID
- Age
- Education
- Sex
- Is_smoking
- Cigs per days
- BPMeds
- Prevalent stroke
- Prevalent Hypertensive
- Diabetes
- Total Cholestrol
- sysBP
- diaBP
- BMI
- Heart Rate
- Glucose
- Ten Year CHD

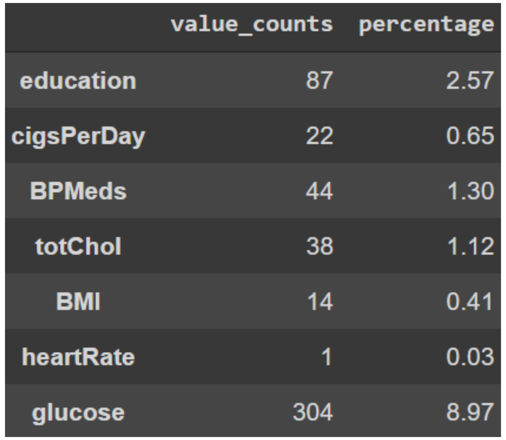
Exploratory Data Analysis :

Null Values

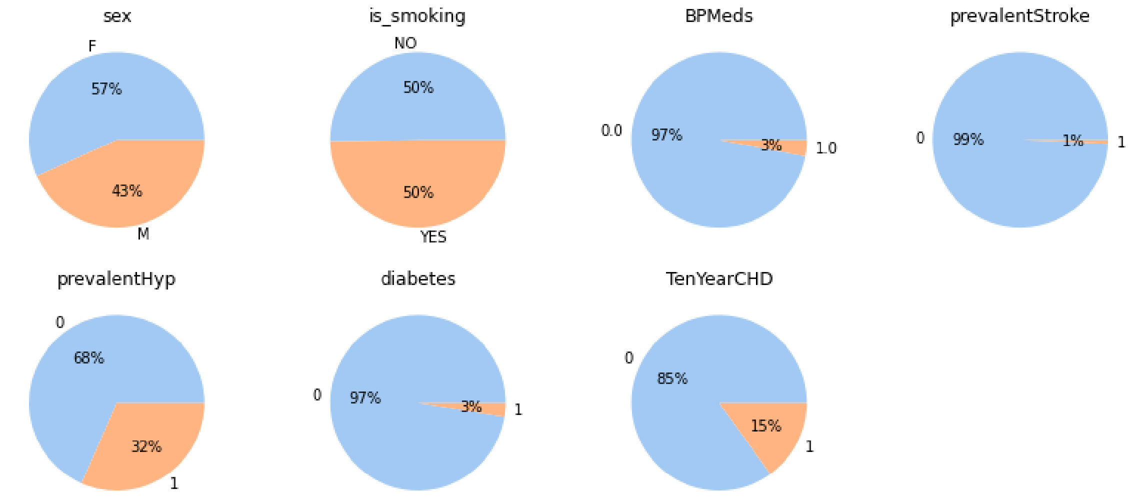
On exploring we see that these variables have null values in them.

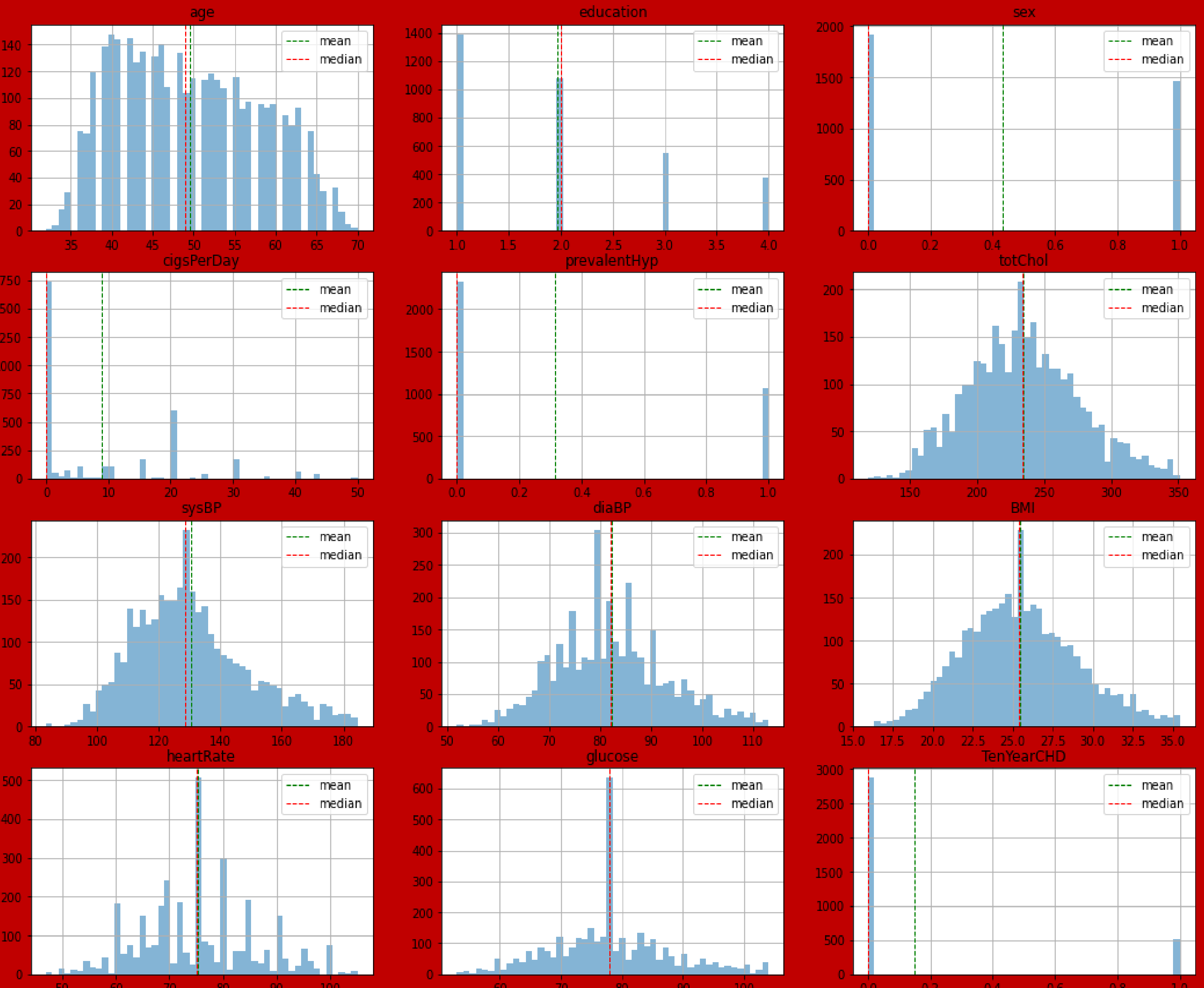






	value_counts	percentage
education	87	2.57
cigsPerDay	22	0.65
BPMeds	44	1.30
totChol	38	1.12
BMI	14	0.41
heartRate	1	0.03
glucose	304	8.97




age
0
20
40
60
80
100
120
140
35
40
45
50
55
60
65
70
mean
median
education
0
200
400
600
800
1000
1200
1400
1.0
1.5
2.0
2.5
3.0
3.5
4.0
mean
median
sex
0
500
1000
1500
2000
0.0
0.2
0.4
0.6
0.8
1.0
mean
median
cigsPerDay
0
250
500
750
1000
1250
1500
1750
0
10
20
30
40
50
mean
median
prevalentHyp
0
500
1000
1500
2000
0.0
0.2
0.4
0.6
0.8
1.0
mean
median
totChol
0
50
100
150
200
150
200
250
300
350
mean
median
sysBP
0
50
100
150
200
80
100
120
140
160
180
mean
median
diaBP
0
50
100
150
200
250
300
50
60
70
80
90
100
110
mean
median
BMI
0
50
100
150
200
15.0
17.5
20.0
22.5
25.0
27.5
30.0
32.5
35.0
mean
median
heartRate
0
100
200
300
400
500
50
60
70
80
90
100
mean
median
glucose
0
100
200
300
400
500
600
60
70
80
90
100
mean
median
TenYearCHD
0
500
1000
1500
2000
2500
3000
0.0
0.2
0.4
0.6
0.8
1.0
mean
median

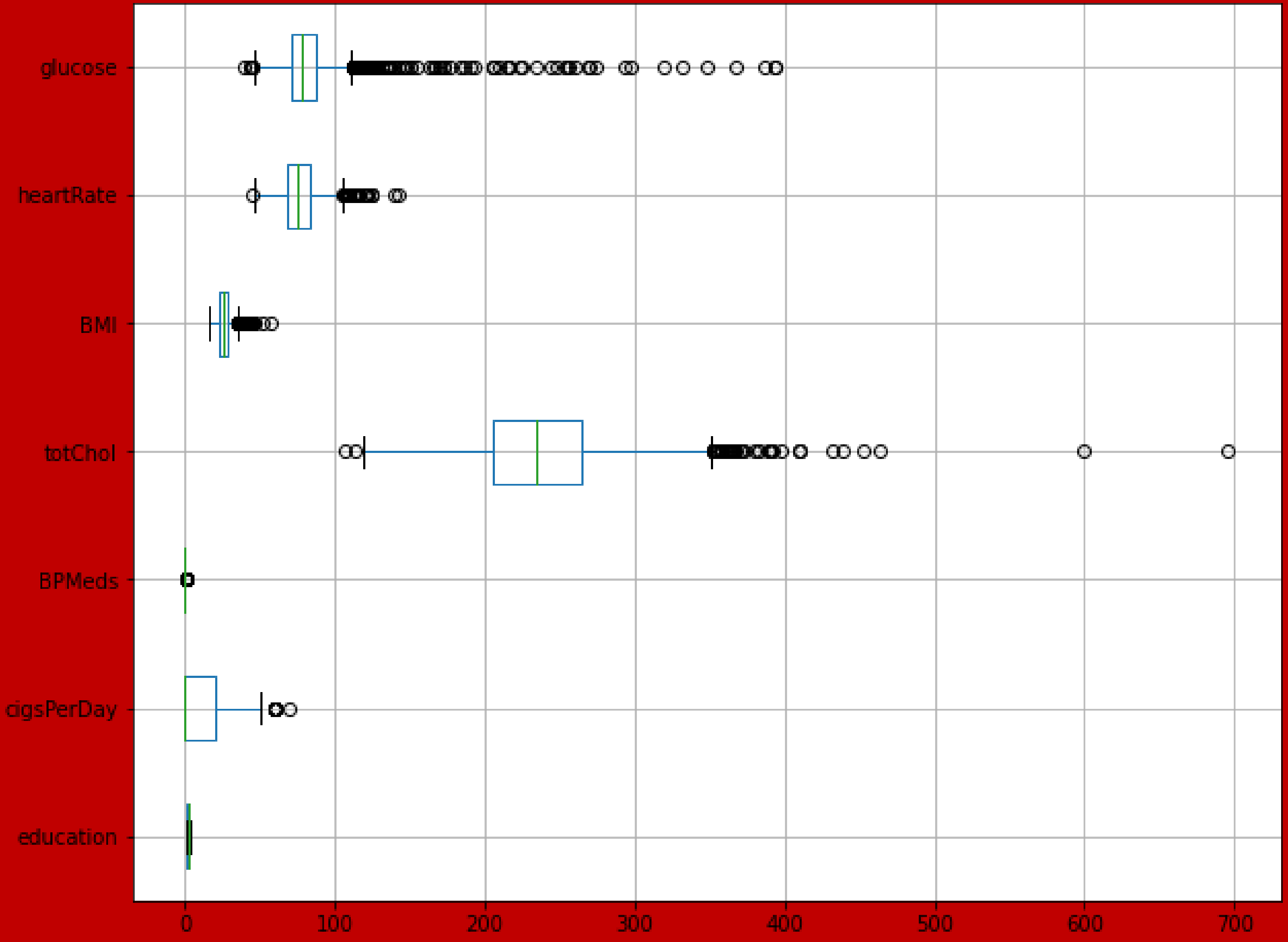
Null - Value treatment

Deciding which value to impute in place of null values.

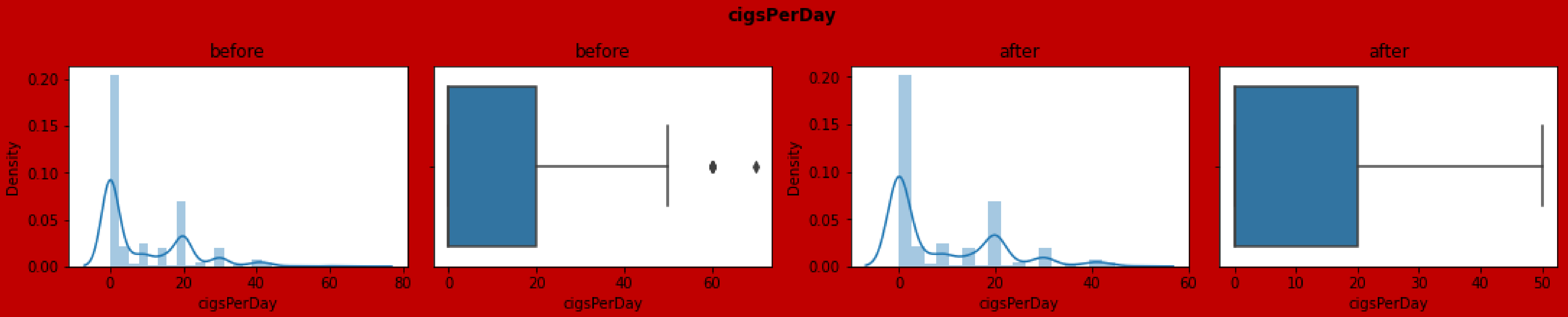
Mean

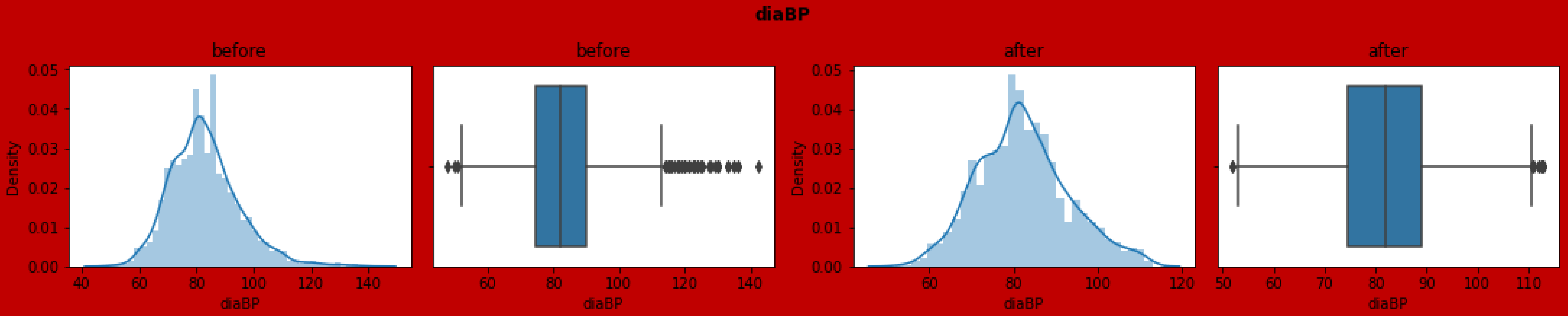
Mode

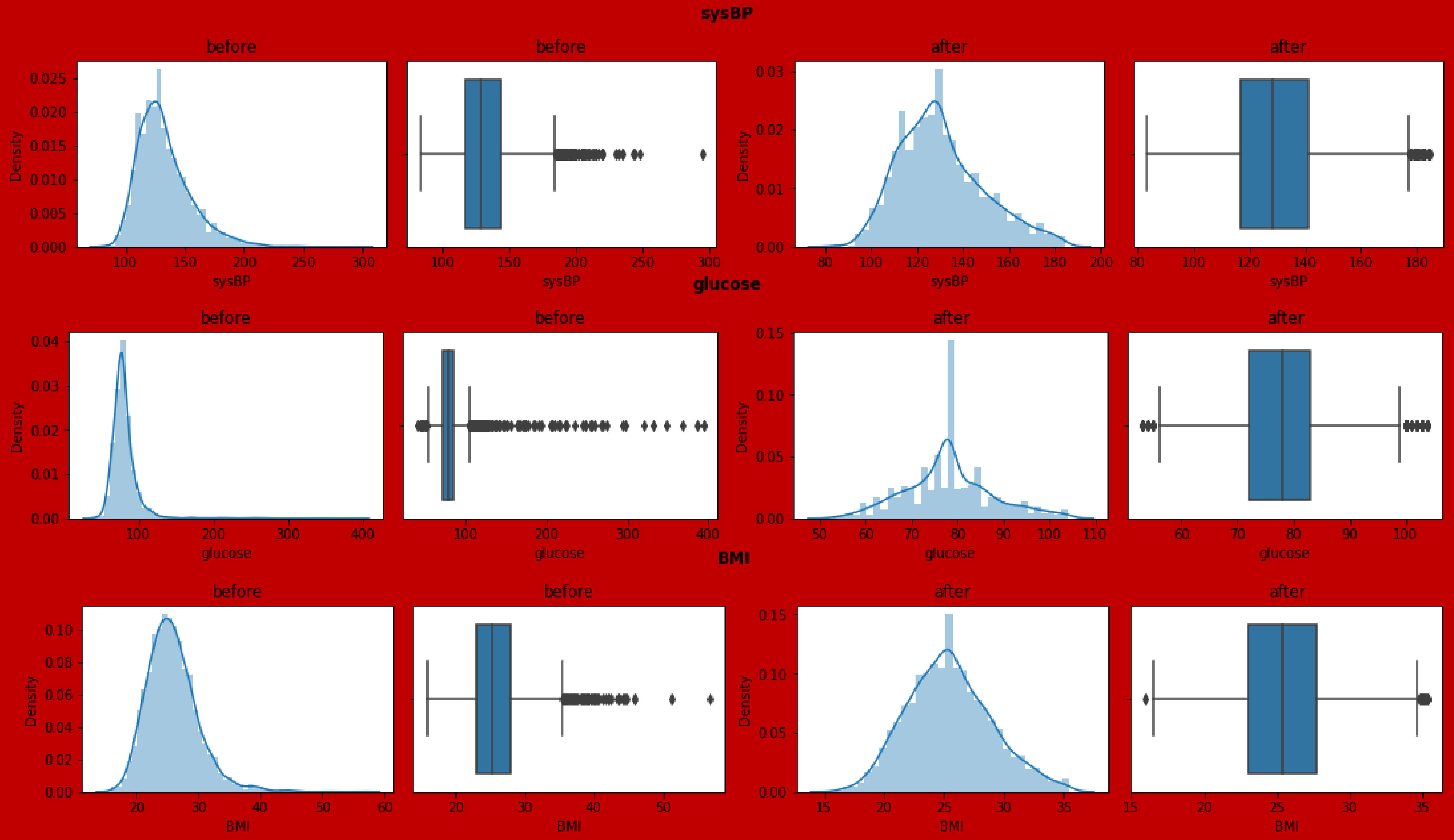
Median

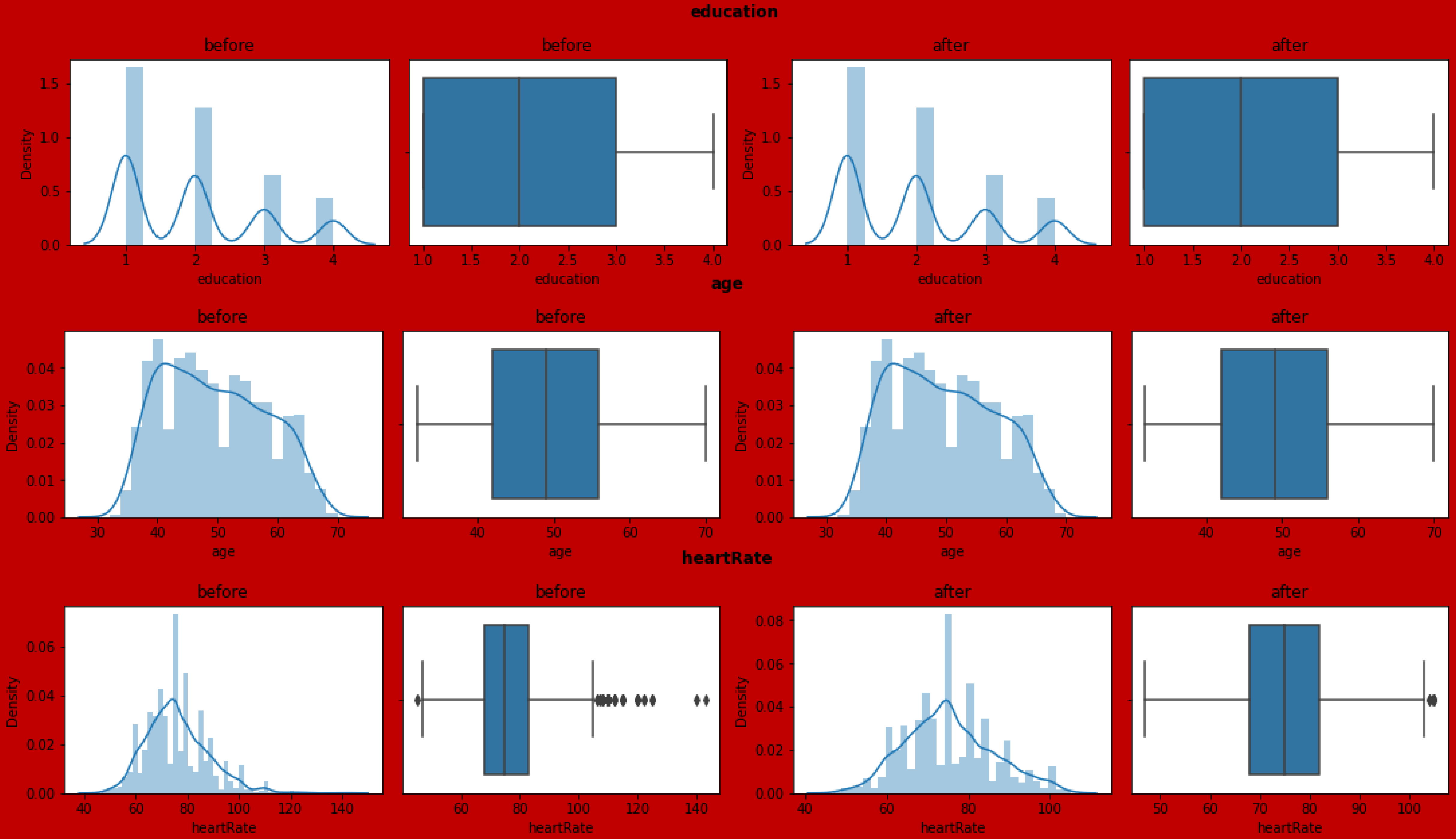


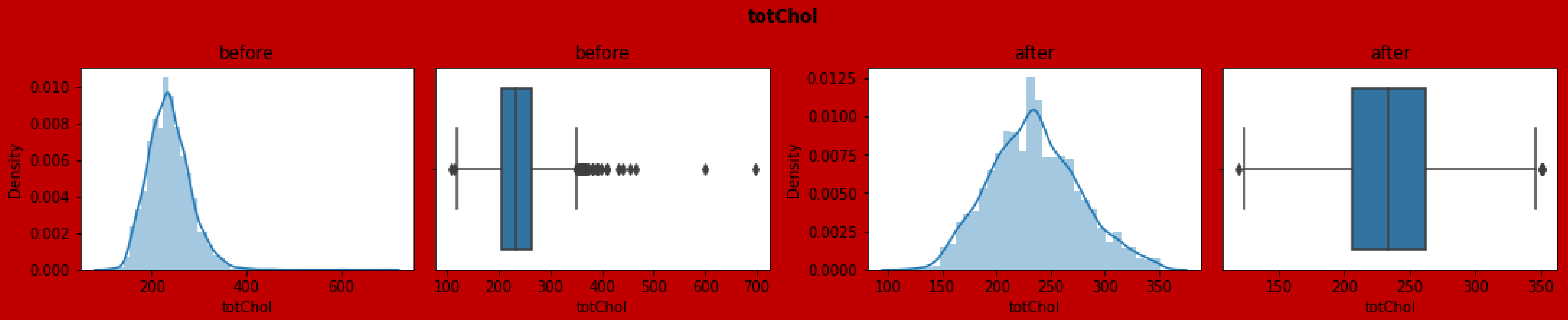
Outlier treatment





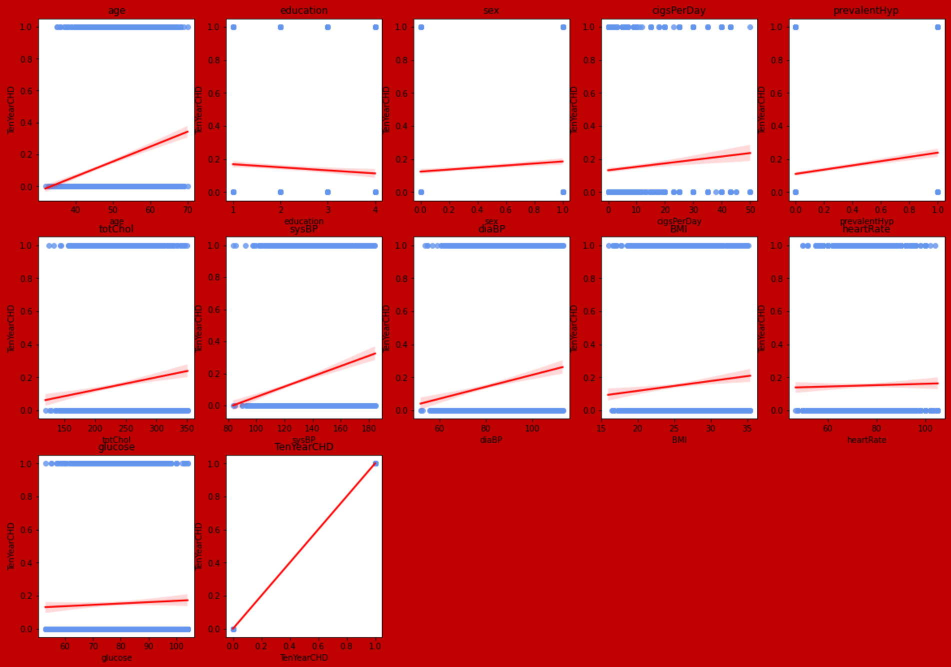

sysBP
before
Density
0.025
0.020
0.015
0.010
0.005
0.000
100
150
200
250
300
sysBP
before
100
150
200
250
300
sysBP
after
Density
0.03
0.02
0.01
0.00
80
100
120
140
160
180
200
sysBP
after
80
100
120
140
160
180
sysBP
glucose
before
Density
0.04
0.03
0.02
0.01
0.00
100
200
300
400
glucose
before
100
200
300
400
glucose
after
Density
0.15
0.10
0.05
0.00
50
60
70
80
90
100
110
glucose
after
60
70
80
90
100
glucose
BMI
before
Density
0.10
0.08
0.06
0.04
0.02
0.00
20
30
40
50
60
BMI
before
20
30
40
50
BMI
after
Density
0.15
0.10
0.05
0.00
15
20
25
30
35
BMI
after
15
20
25
30
35
BMI


education
before
before
after
after
Density
0.0
0.5
1.0
1.5
1
2
3
4
1.0
1.5
2.0
2.5
3.0
3.5
4.0
education
age
before
before
after
after
Density
0.00
0.01
0.02
0.03
0.04
30
40
50
60
70
age
heartRate
before
before
after
after
Density
0.00
0.02
0.04
0.06
0.08
40
50
60
70
80
90
100
120
140
heartRate

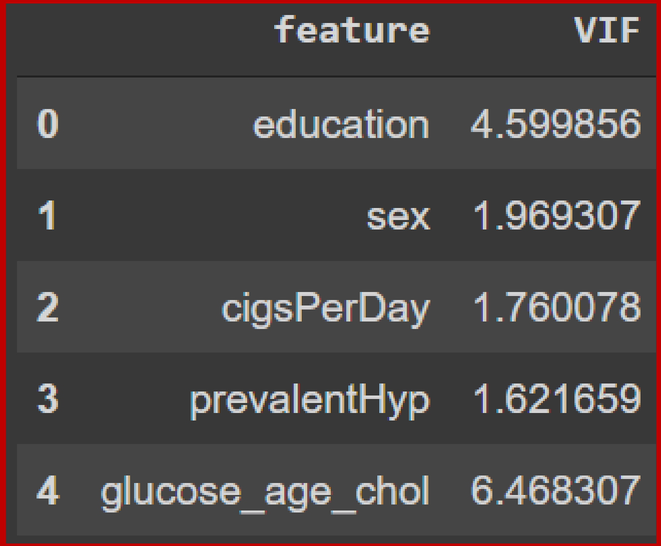

totChol
before
before
after
after
Density
0.010
0.008
0.006
0.004
0.002
0.000
200
400
600
totChol
100
200
300
400
500
600
700
totChol
Density
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
100
150
200
250
300
350
totChol
150
200
250
300
350
totChol

Logistic Regression Data Preparation

Verifying linear dependency



Removing Multicollinearity

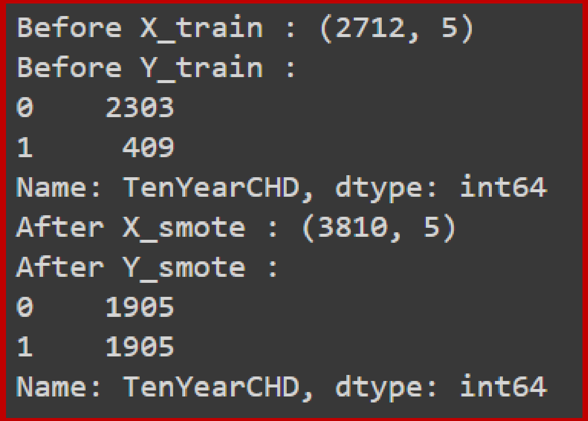


	feature	VIF
0	education	4.599856
1	sex	1.969307
2	cigsPerDay	1.760078
3	prevalentHyp	1.621659
4	glucose_age_chol	6.468307

Preparing data for SVC*s*

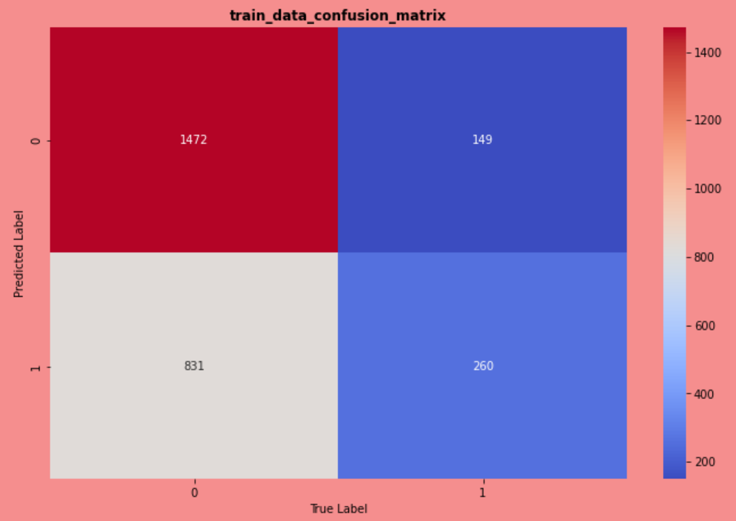
Preparing data for Ensemble tree models.

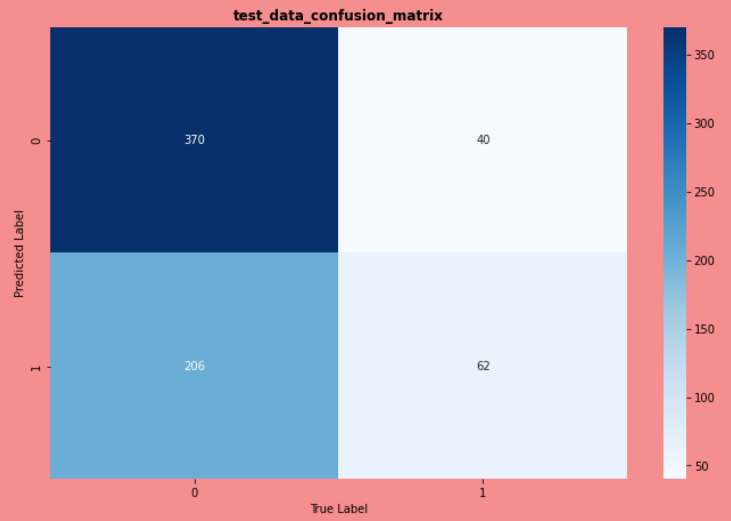
Handling Data Imbalance.

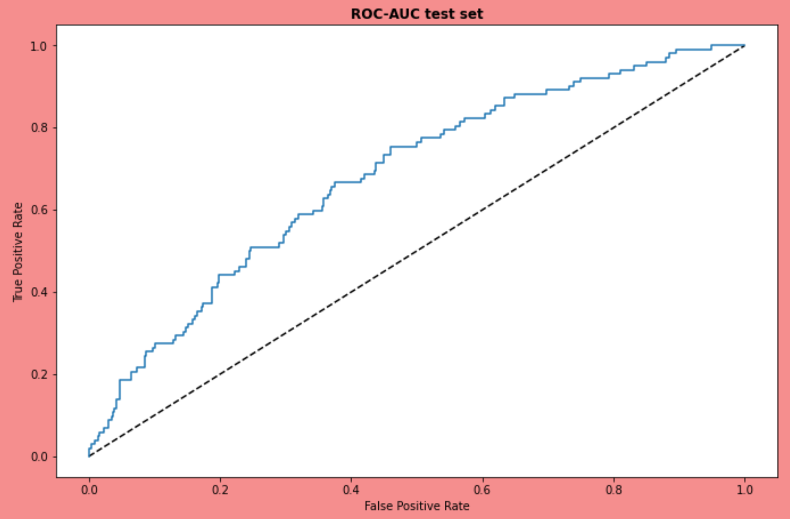


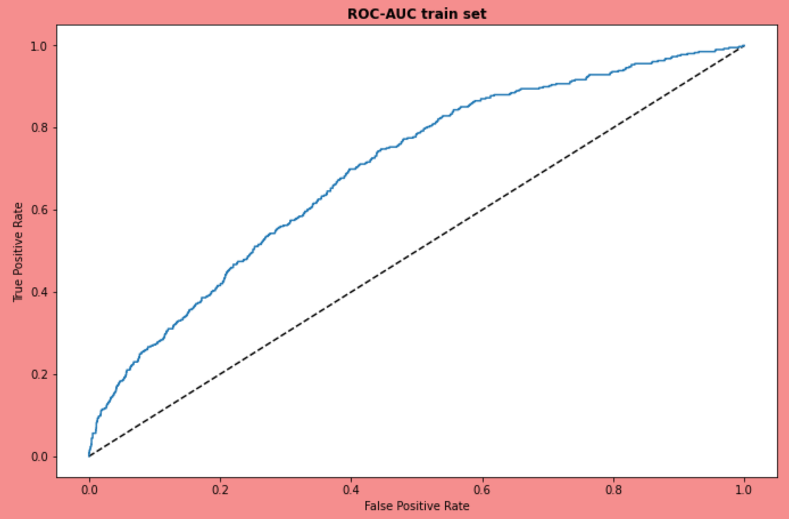
```
Before X_train : (2712, 5)
Before Y_train :
0    2303
1     409
Name: TenYearCHD, dtype: int64
After X_smote : (3810, 5)
After Y_smote :
0    1905
1    1905
Name: TenYearCHD, dtype: int64
```

Logistic Regression Implementation

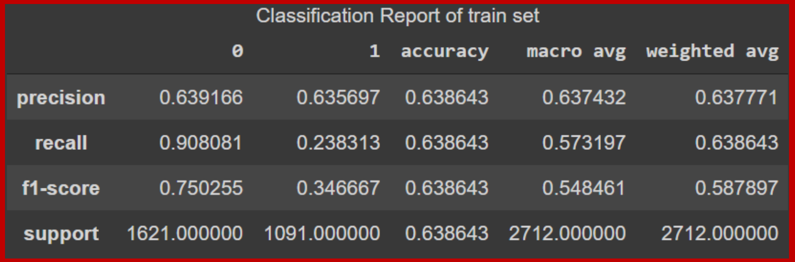






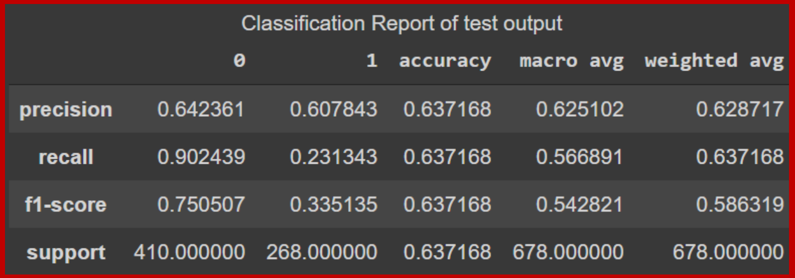


Logistic Regression Implementation (Contd..)



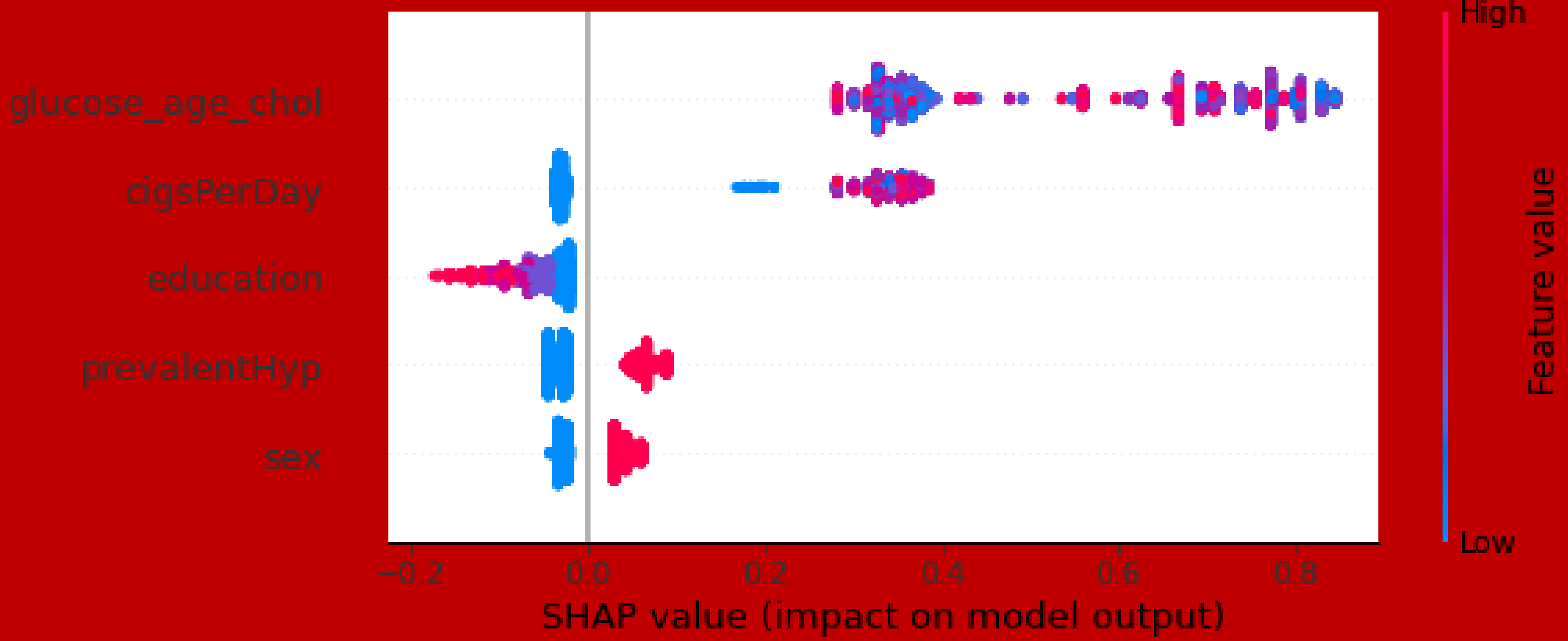
Classification Report of train set

	0	1	accuracy	macro avg	weighted avg
precision	0.639166	0.635697	0.638643	0.637432	0.637771
recall	0.908081	0.238313	0.638643	0.573197	0.638643
f1-score	0.750255	0.346667	0.638643	0.548461	0.587897
support	1621.000000	1091.000000	0.638643	2712.000000	2712.000000

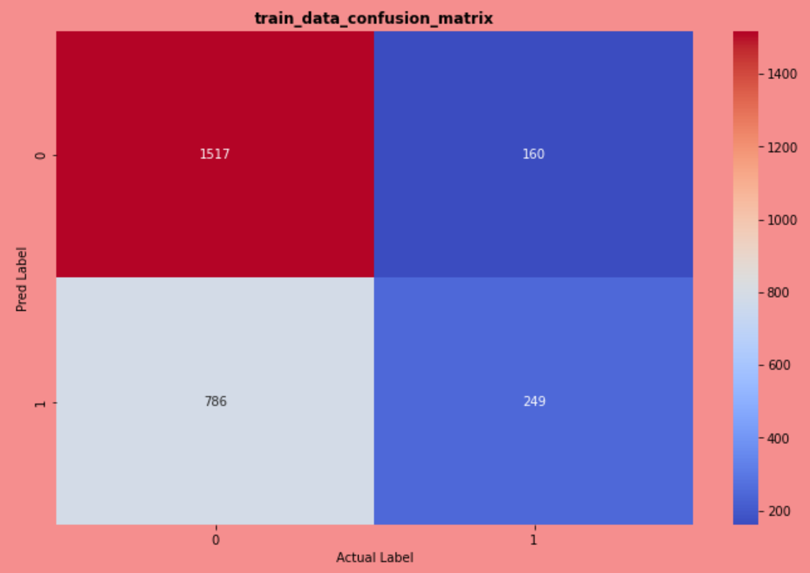


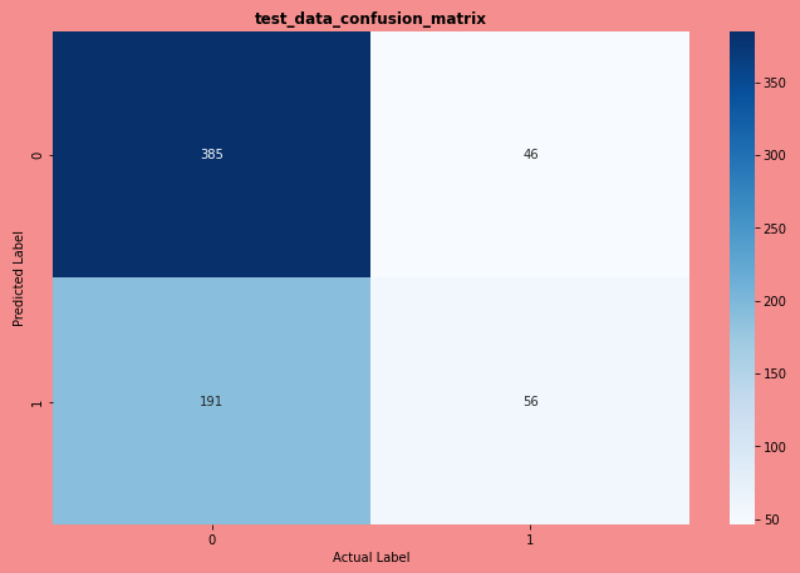
Classification Report of test output

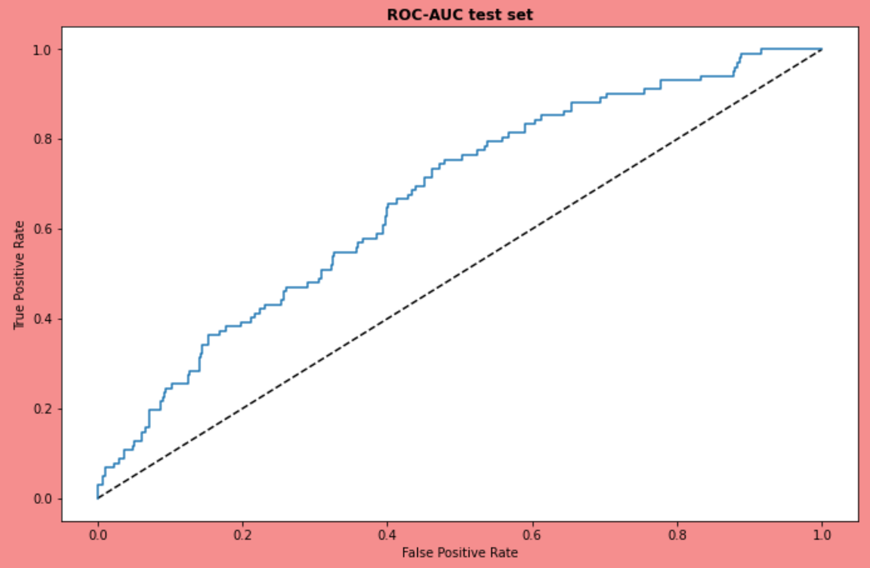
	0	1	accuracy	macro avg	weighted avg
precision	0.642361	0.607843	0.637168	0.625102	0.628717
recall	0.902439	0.231343	0.637168	0.566891	0.637168
f1-score	0.750507	0.335135	0.637168	0.542821	0.586319
support	410.000000	268.000000	0.637168	678.000000	678.000000

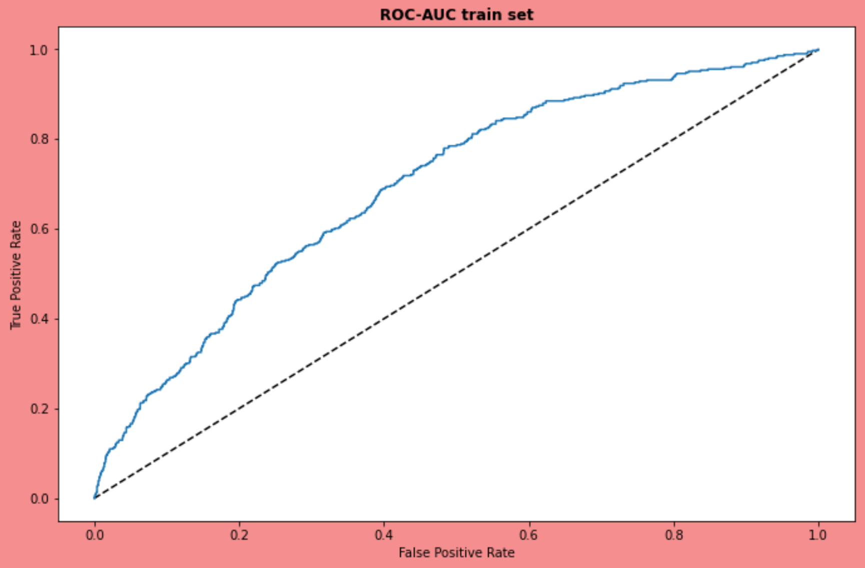


Gaussian Naïve Bayes Implementation



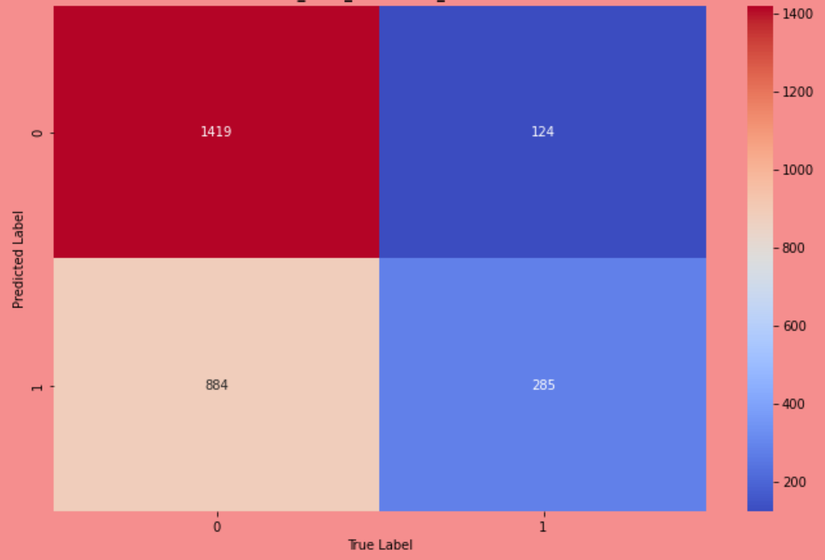




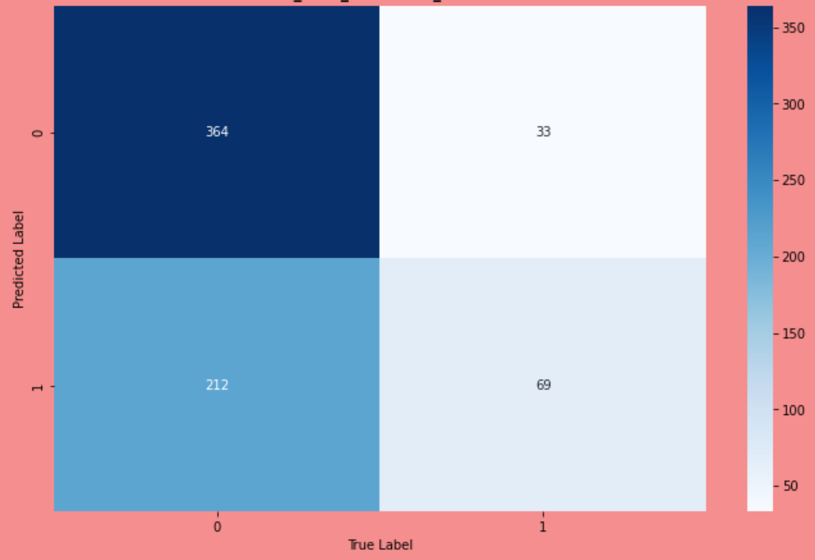


Support Vector Classifier Implementation

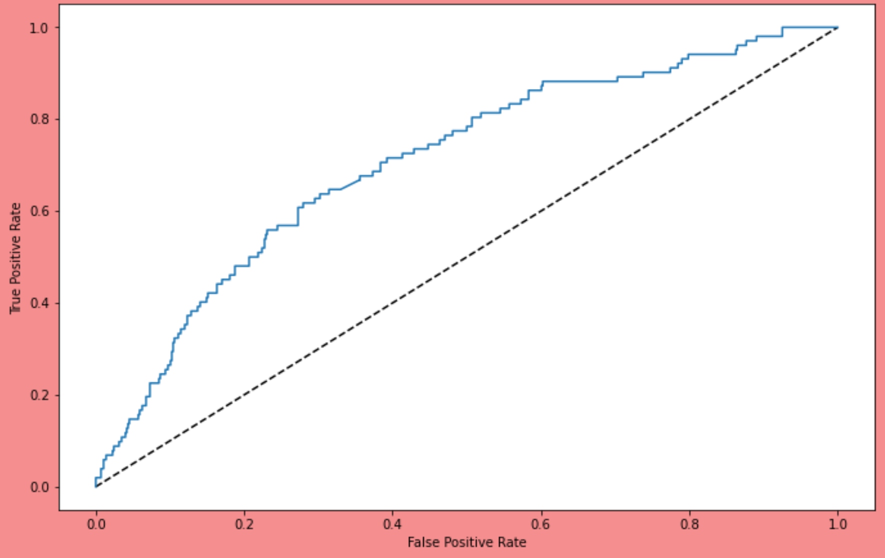
train_data_confusion_matrix



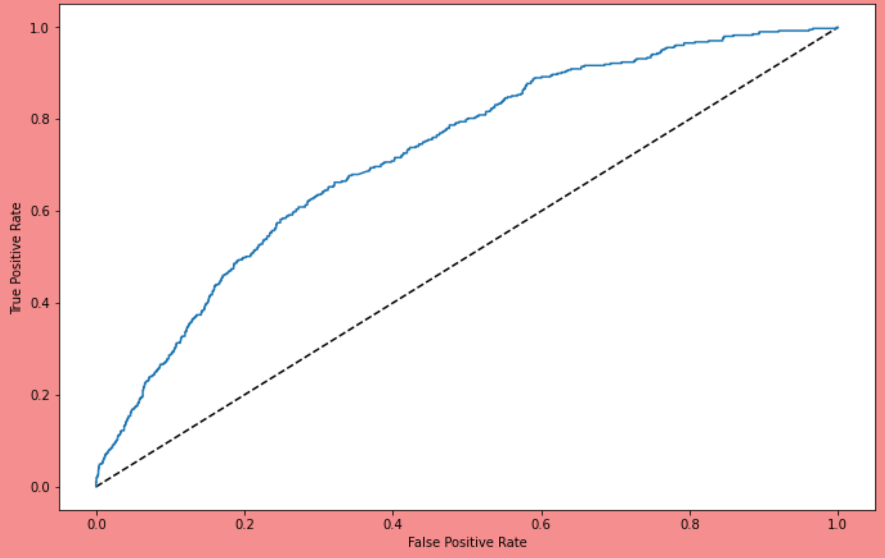
test_data_confusion_matrix



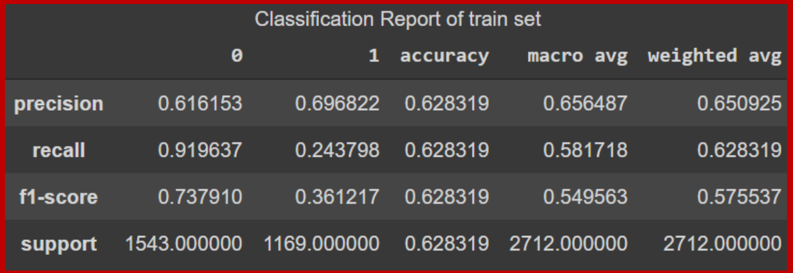
ROC-AUC test set



ROC-AUC train set

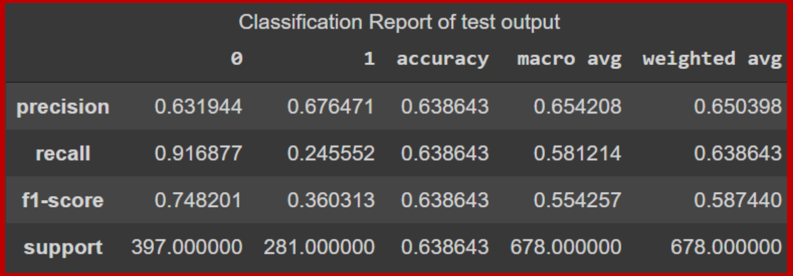


Support Vector Classifier Implementation (Contd..)



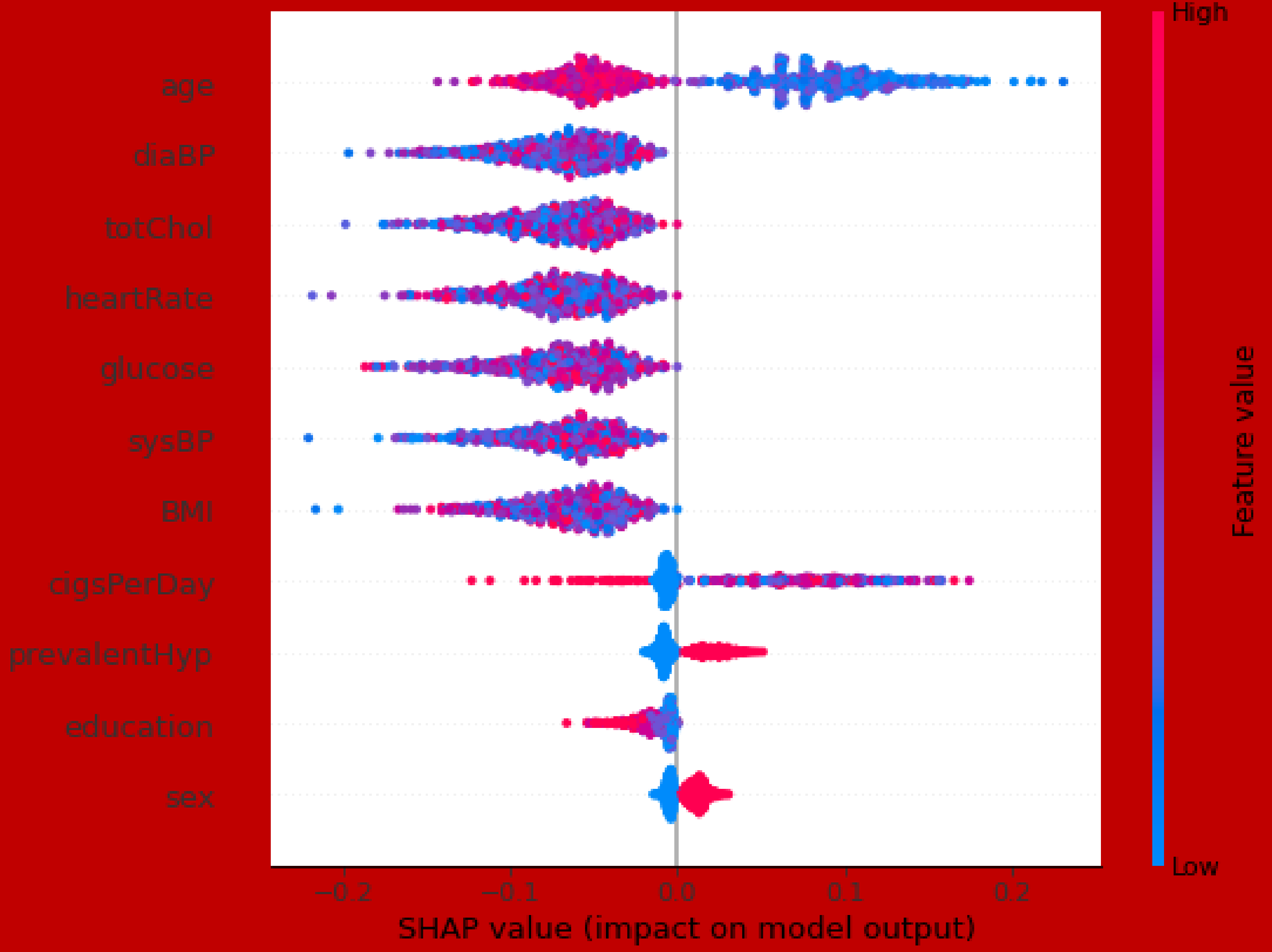
Classification Report of train set

	0	1	accuracy	macro avg	weighted avg
precision	0.616153	0.696822	0.628319	0.656487	0.650925
recall	0.919637	0.243798	0.628319	0.581718	0.628319
f1-score	0.737910	0.361217	0.628319	0.549563	0.575537
support	1543.000000	1169.000000	0.628319	2712.000000	2712.000000

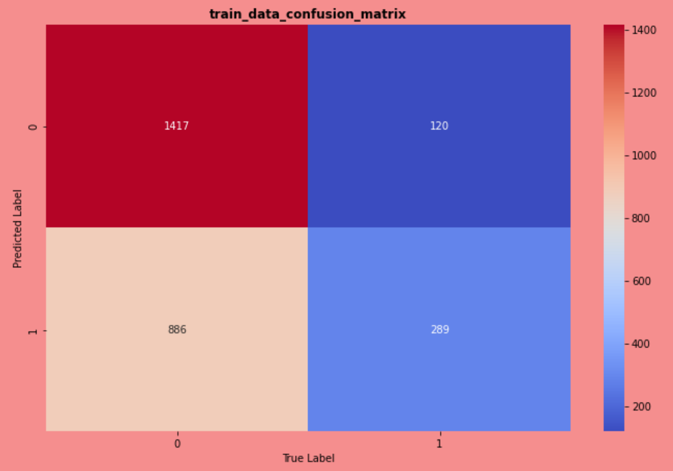


Classification Report of test output

	0	1	accuracy	macro avg	weighted avg
precision	0.631944	0.676471	0.638643	0.654208	0.650398
recall	0.916877	0.245552	0.638643	0.581214	0.638643
f1-score	0.748201	0.360313	0.638643	0.554257	0.587440
support	397.000000	281.000000	0.638643	678.000000	678.000000

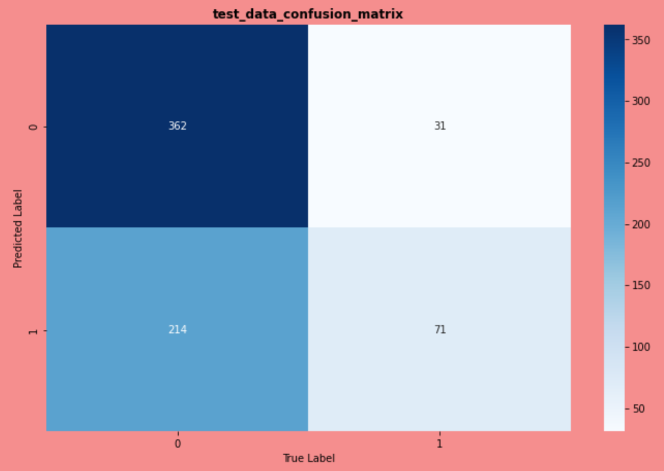


Support Vector Classifier with balanced loss function Implementation



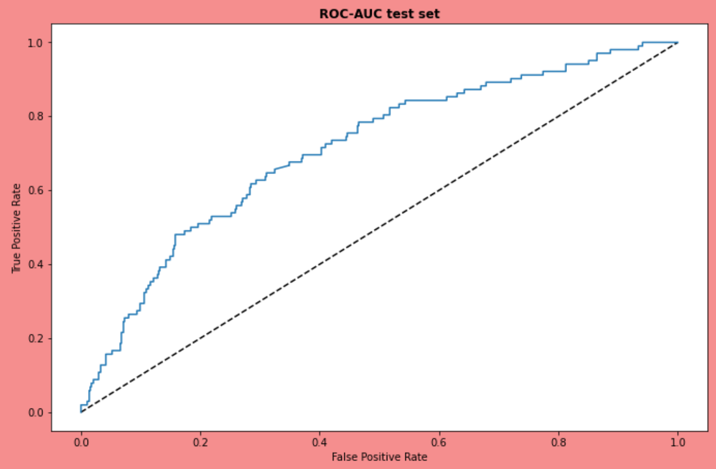
train_data_confusion_matrix

Predicted Label \ True Label	0	1
0	1417	120
1	886	289

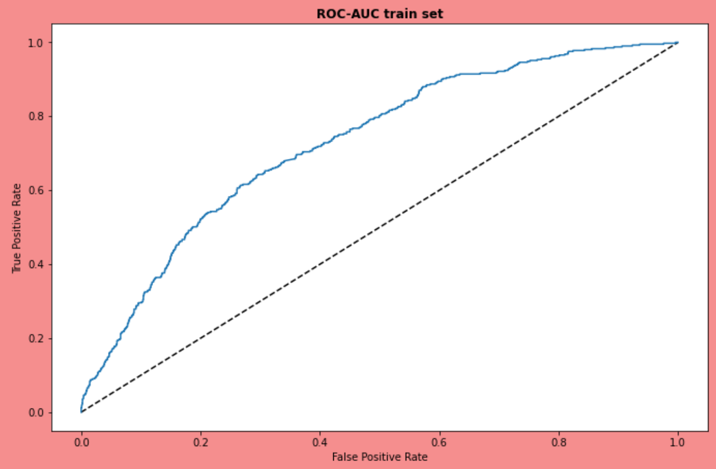


test_data_confusion_matrix

Predicted Label \ True Label	0	1
0	362	31
1	214	71

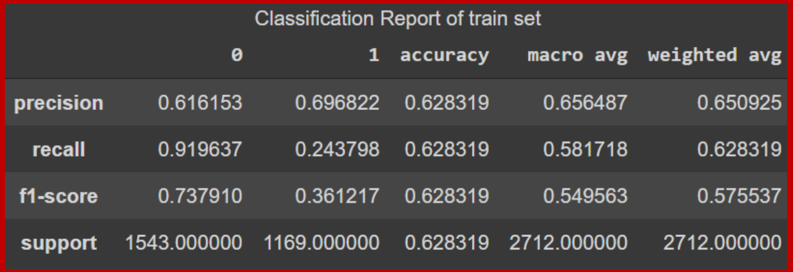


ROC-AUC test set



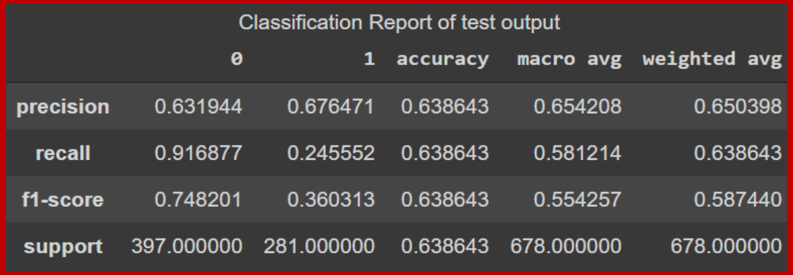
ROC-AUC train set

Support Vector Classifier with balanced loss Implementation (Contd..)



Classification Report of train set

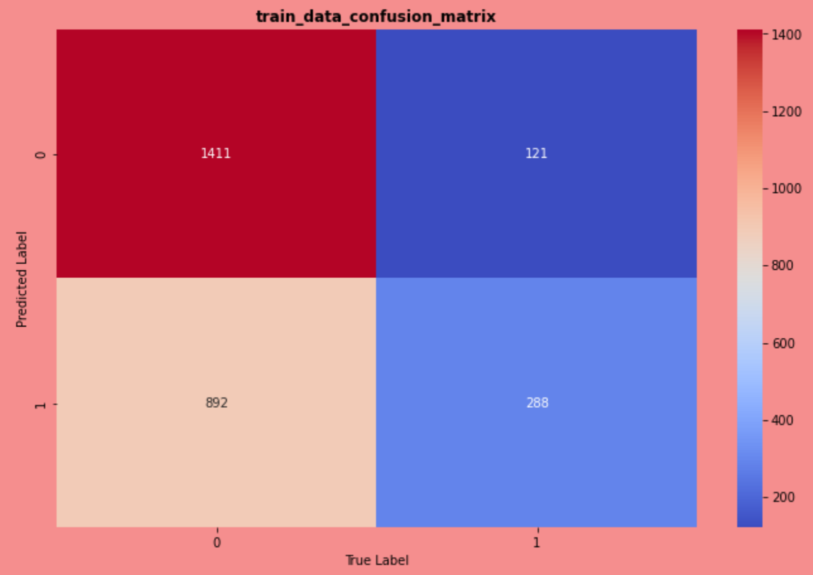
	0	1	accuracy	macro avg	weighted avg
precision	0.616153	0.696822	0.628319	0.656487	0.650925
recall	0.919637	0.243798	0.628319	0.581718	0.628319
f1-score	0.737910	0.361217	0.628319	0.549563	0.575537
support	1543.000000	1169.000000	0.628319	2712.000000	2712.000000

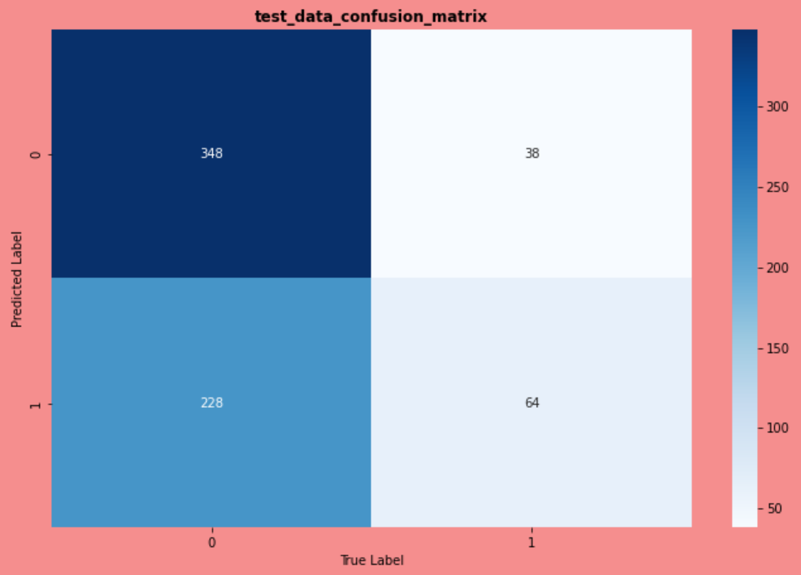


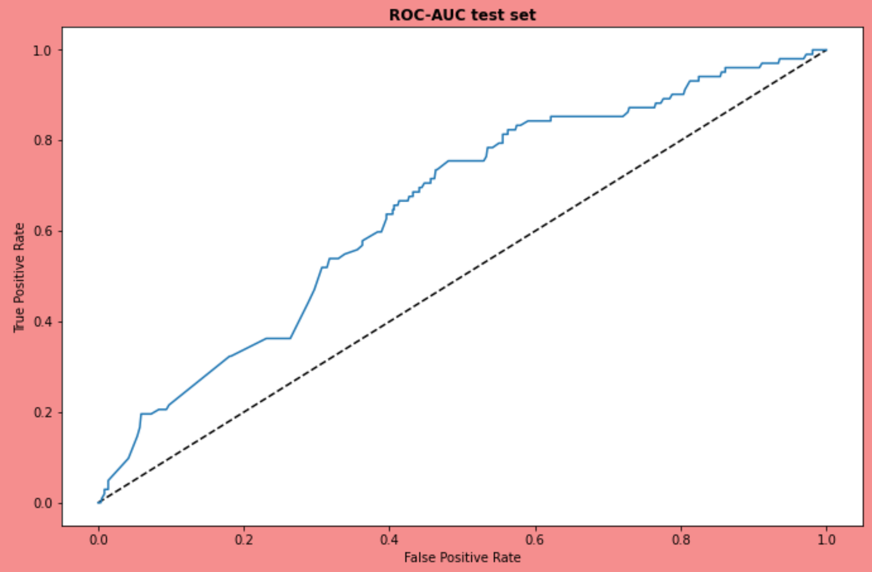
Classification Report of test output

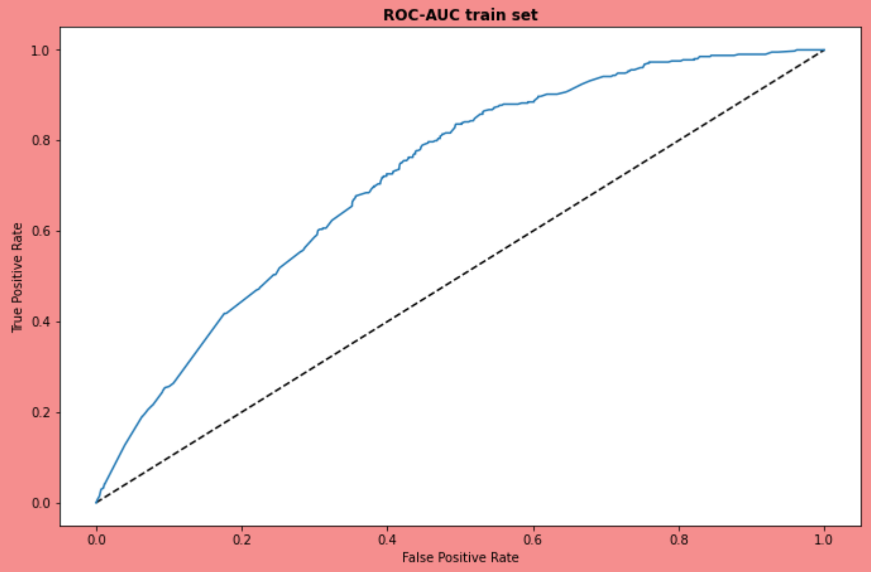
	0	1	accuracy	macro avg	weighted avg
precision	0.631944	0.676471	0.638643	0.654208	0.650398
recall	0.916877	0.245552	0.638643	0.581214	0.638643
f1-score	0.748201	0.360313	0.638643	0.554257	0.587440
support	397.000000	281.000000	0.638643	678.000000	678.000000

LGBM Implementation

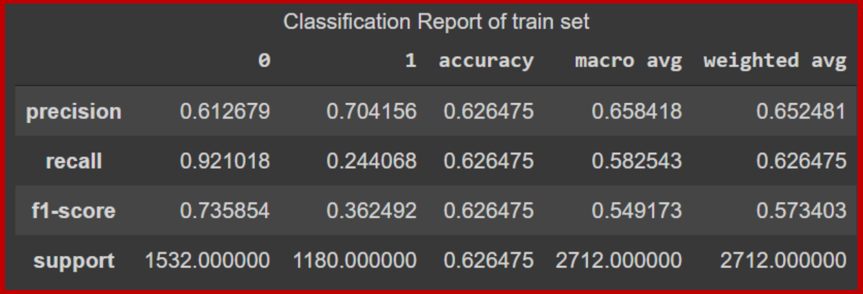






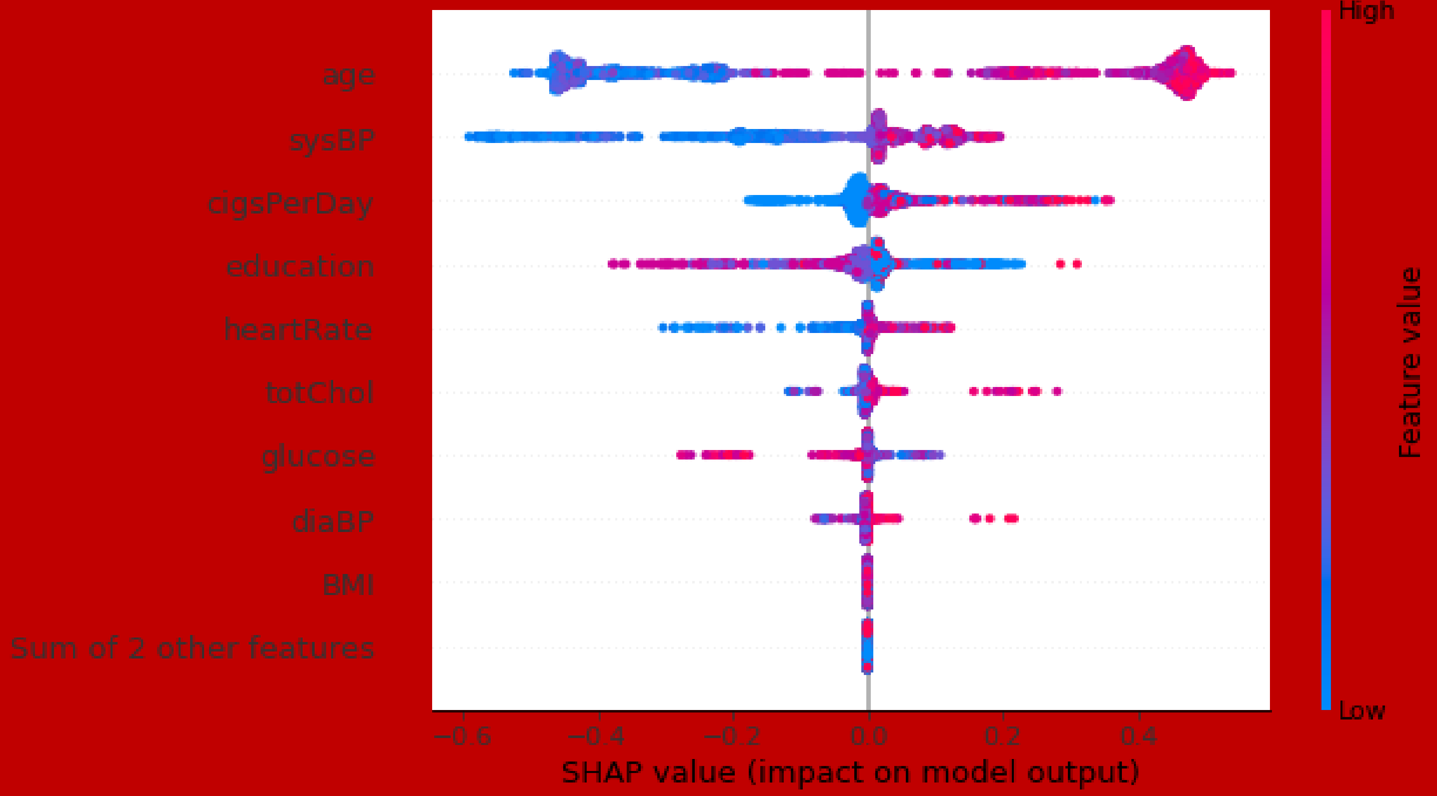


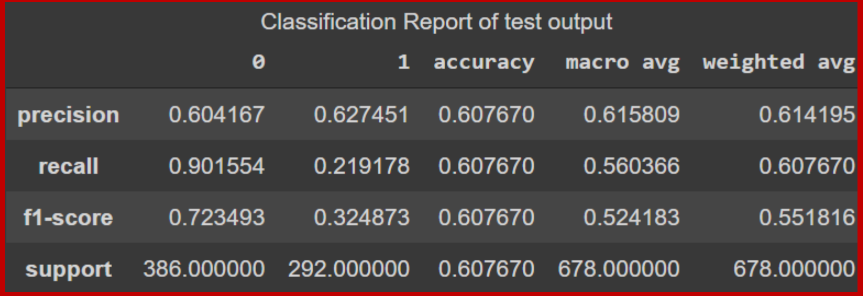
LGBM Implementation



Classification Report of train set

	0	1	accuracy	macro avg	weighted avg
precision	0.612679	0.704156	0.626475	0.658418	0.652481
recall	0.921018	0.244068	0.626475	0.582543	0.626475
f1-score	0.735854	0.362492	0.626475	0.549173	0.573403
support	1532.000000	1180.000000	0.626475	2712.000000	2712.000000

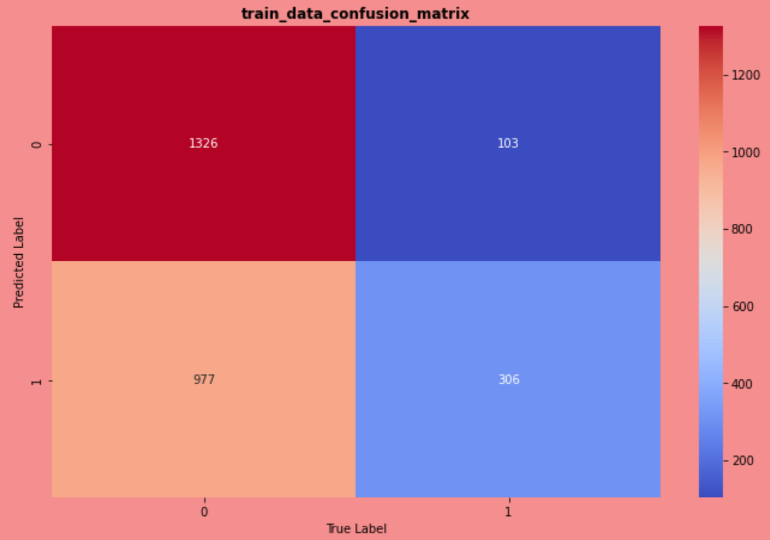




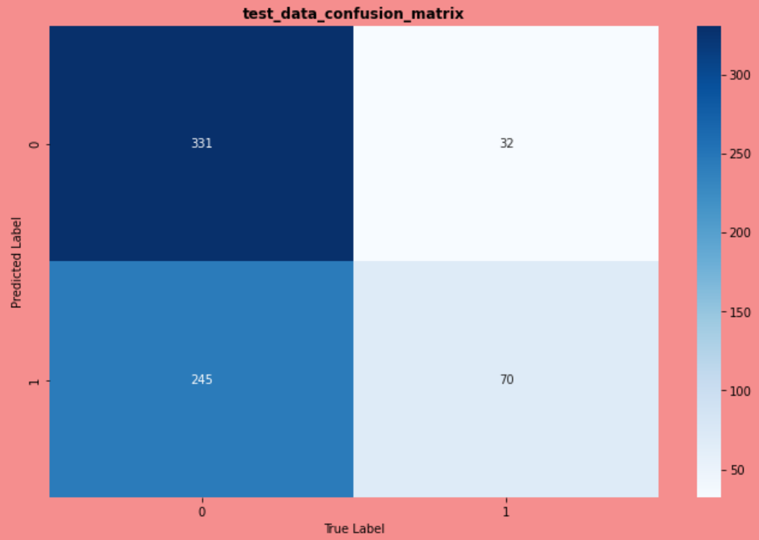
Classification Report of test output

	0	1	accuracy	macro avg	weighted avg
precision	0.604167	0.627451	0.607670	0.615809	0.614195
recall	0.901554	0.219178	0.607670	0.560366	0.607670
f1-score	0.723493	0.324873	0.607670	0.524183	0.551816
support	386.000000	292.000000	0.607670	678.000000	678.000000

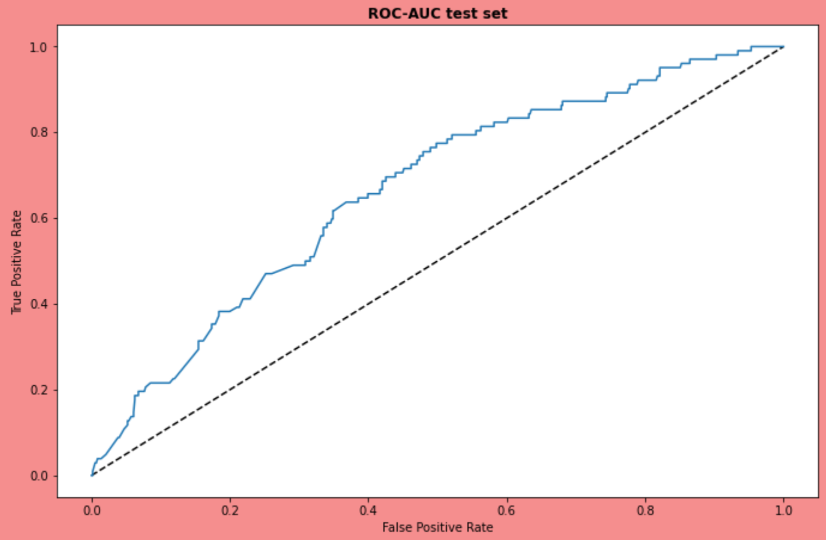
XGRFB Implementation



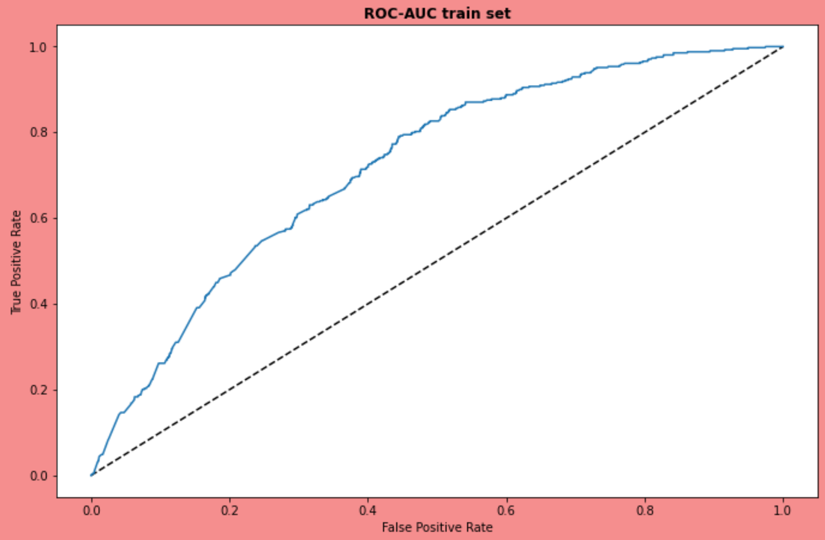
train_data_confusion_matrix



test_data_confusion_matrix

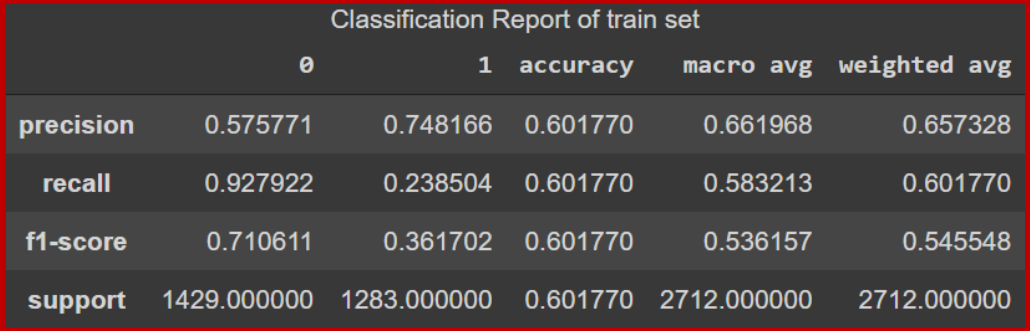


ROC-AUC test set



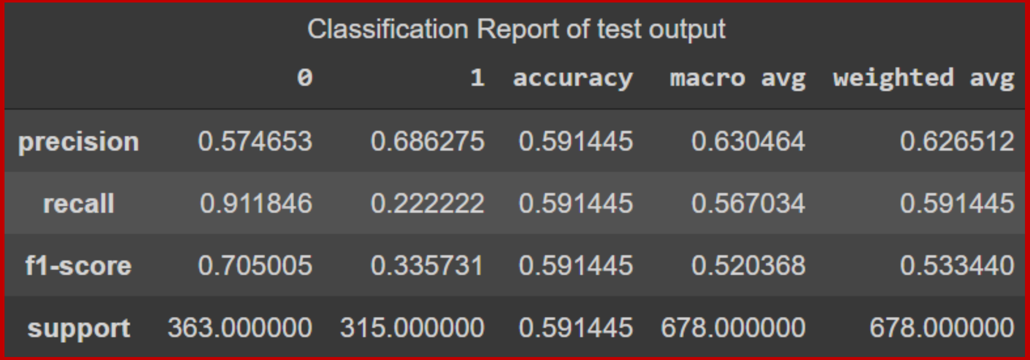
ROC-AUC train set

XGRFB Implementation



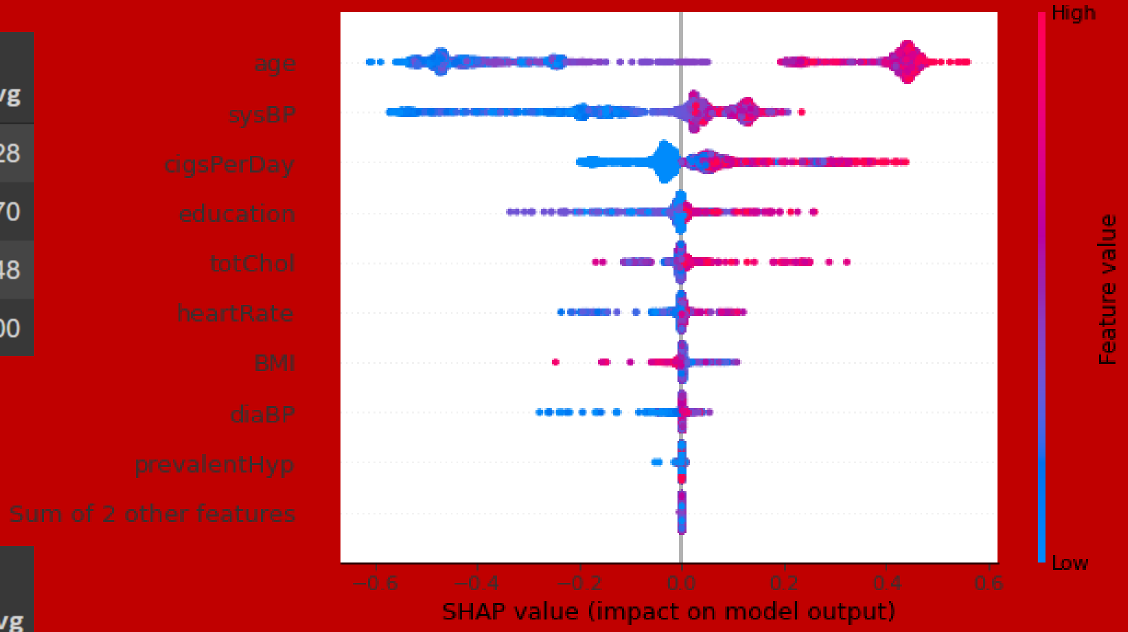
Classification Report of train set

	0	1	accuracy	macro avg	weighted avg
precision	0.575771	0.748166	0.601770	0.661968	0.657328
recall	0.927922	0.238504	0.601770	0.583213	0.601770
f1-score	0.710611	0.361702	0.601770	0.536157	0.545548
support	1429.000000	1283.000000	0.601770	2712.000000	2712.000000

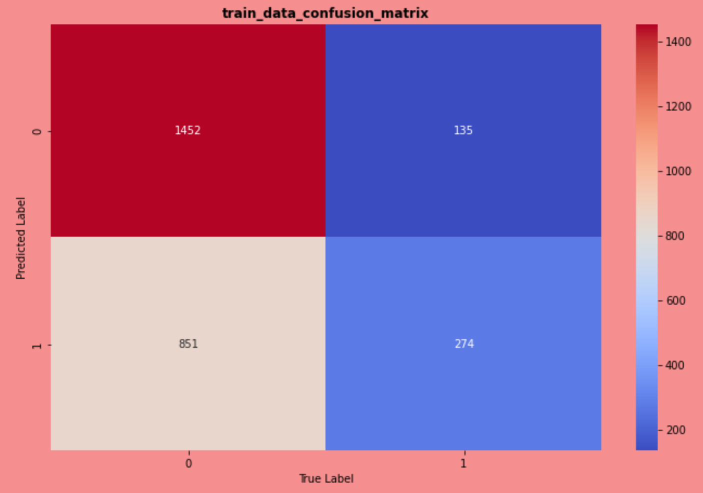


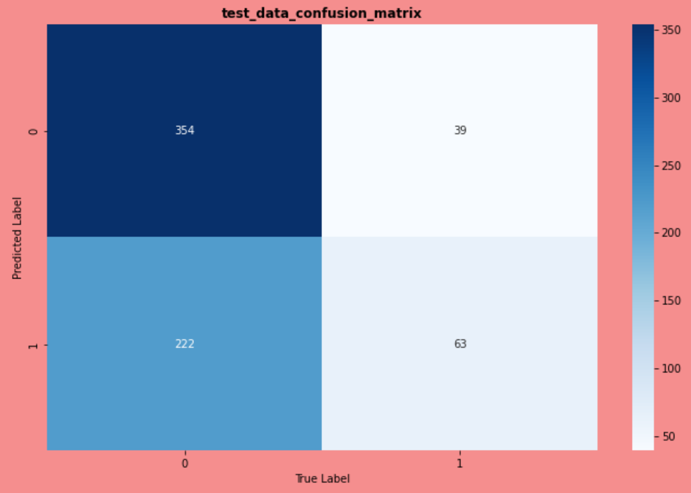
Classification Report of test output

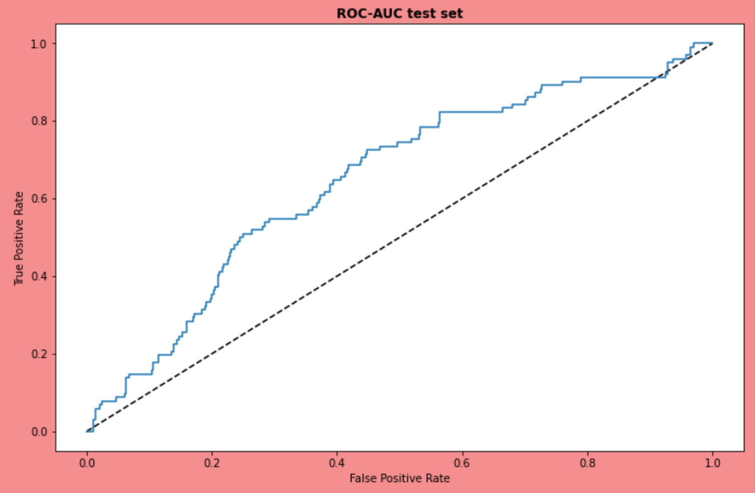
	0	1	accuracy	macro avg	weighted avg
precision	0.574653	0.686275	0.591445	0.630464	0.626512
recall	0.911846	0.222222	0.591445	0.567034	0.591445
f1-score	0.705005	0.335731	0.591445	0.520368	0.533440
support	363.000000	315.000000	0.591445	678.000000	678.000000

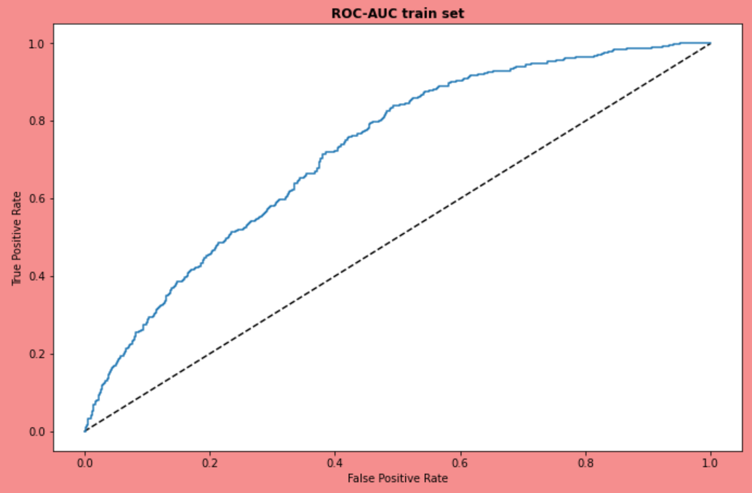


Random Forest Implementation

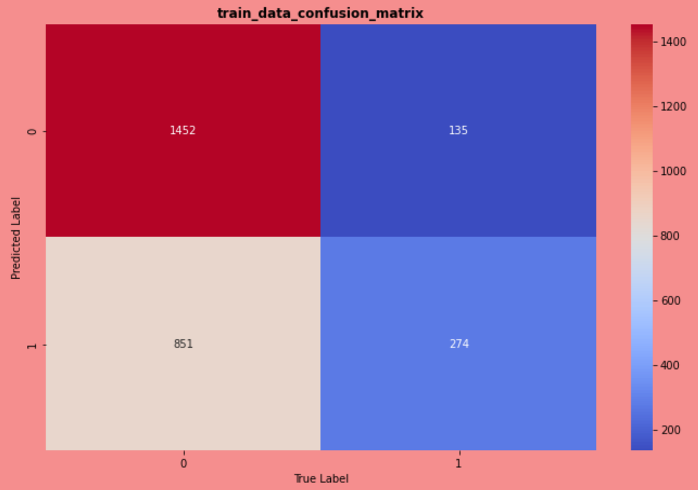


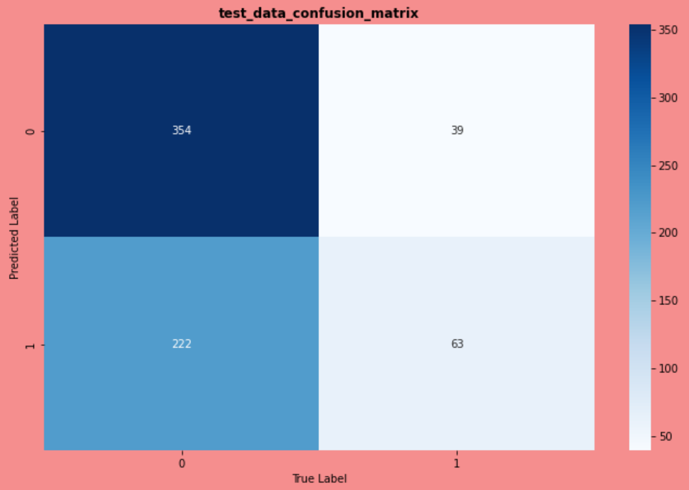


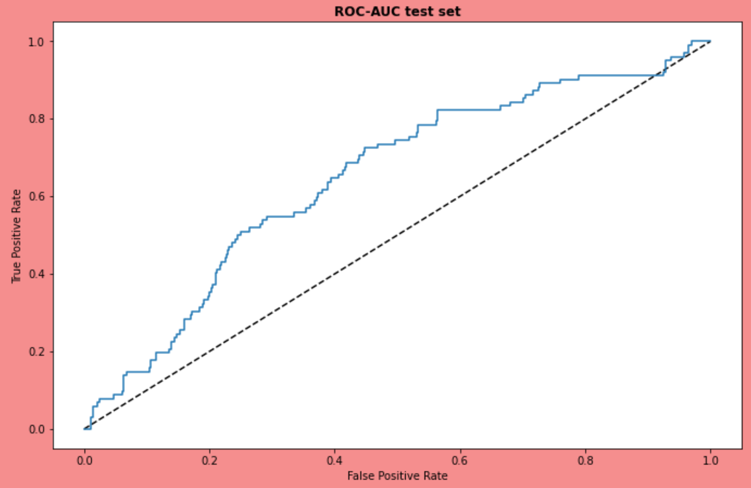


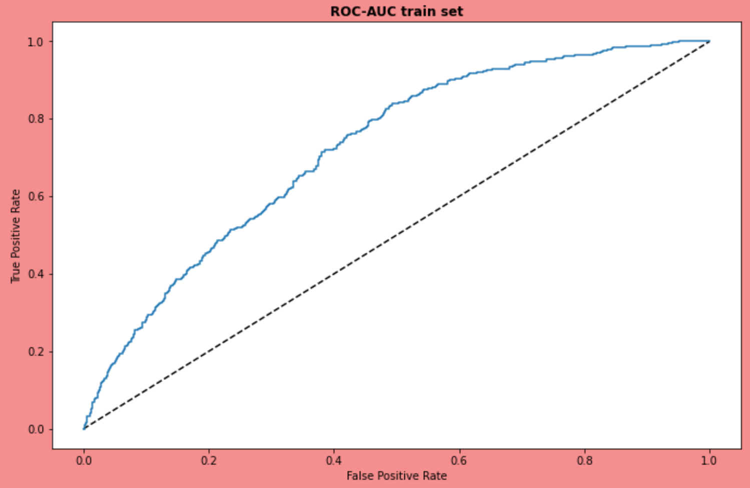


Neural Network Implementation


train_data_confusion_matrix


test_data_confusion_matrix


ROC-AUC test set


ROC-AUC train set

Conclusion..

1. Logistic Regression can predict 61% of the negative values long with 35% of False Negative predictions.

2. Gaussian Naive Bayes can predict 55% of the negative values along with 34% of False Negative predictions.

3. Support Vector Classifier *(without balanced loss function)* can predict 68% of the negative values with 37% of False Negative predictions.

4. Support Vector Classifier *(with balanced loss function)* can predict 70% of the negative values with 37% of False Negative predictions.

5. LGBM can predict 63% of the negative values with 40% of False Negative predictions.

6. XGRFB can predict 69% of the negative values with 42% of False Negative predictions.

7. Random Forest Model can predict 66% of the negative values with 39 % False Negative predictions.