Cardiovascular Risk Prediction

(Technical documentation)

Satyajit Sahoo

Data science trainee,

AlmaBetter, Bangalore

1. Abstract:

We classify binaries as representing the case of whether a person will develop heart problems in the next ten years or not. We implement various Machine Learning models to predict the case.

2. Introduction:

The given dataset is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. Our goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD).

The dataset provides the patients' information. It includes over 4,000 records and 15 attributes.

Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors. Most of the variables are self-explanatory.

The variables involved are as follows:

Demographic:

• **Sex**: male or female("M" or "F")

• **Age**: Age of the patient;(Continuous - Although the recorded ages are whole numbers as they are only years, the concept of age is continuous)

• **Education:** Ordinal, high values represents highly educated.

Behavioral:

• **is_smoking**: whether or not the patient is a current smoker ("YES" or "NO")

• **Cigs Per Day**: the number of cigarettes that the person smoked on average in one day.(can be considered continuous as one can have any number of cigarettes, even half a cigarette.)

Medical History:

• **BP Meds**: whether or not the patient was on blood pressure medication (Nominal)

• **Prevalent Stroke**: whether or not the patient had previously had a stroke (Nominal)

• **Prevalent Hyp**: whether or not the patient was hypertensive (Nominal)

• **Diabetes**: whether or not the patient had diabetes (Nominal)

Current Medical Conditions:

• **Tot Chol**: total cholesterol level (Continuous)

• **Sys BP**: systolic blood pressure (Continuous)

• **Dia BP**: diastolic blood pressure (Continuous)

• **BMI**: Body Mass Index (Continuous)

• **Heart Rate**: Continuous, in medical research, variables such as heart rate though in fact discrete, are considered continuous because of a large number of possible values.

• **Glucose**: The glucose level in blood.

• **TenYearCHD**: Abbreviation for Ten Year coronary heart disease, nominal and our target variable as well.

3. EDA:

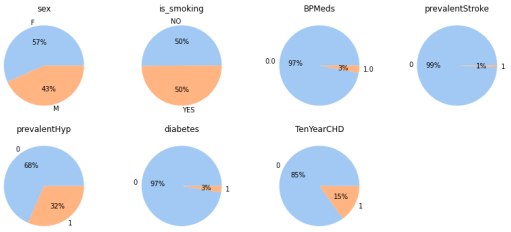
On exploring the dataset we see that some variables have null values.

	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

So, we have to deal with these null values further in the process.

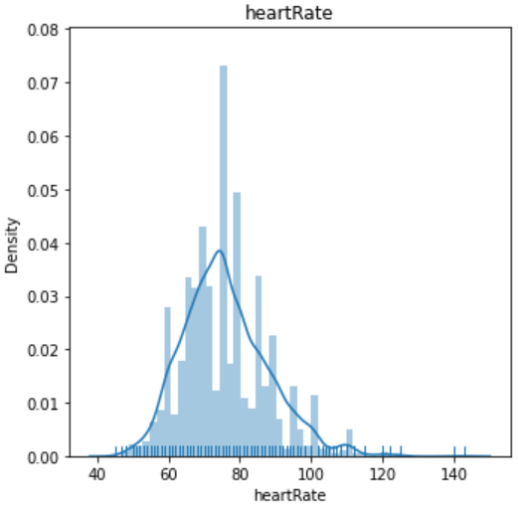
Next, we divide the variables into numerical and categorical variables.

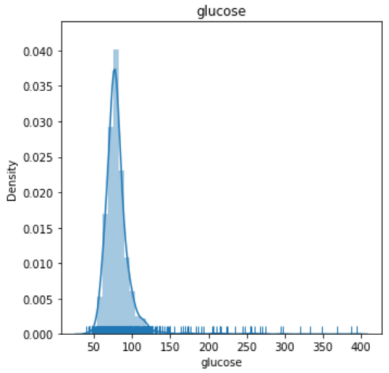
For the categorical variables, we see that the target variable TenYearCHD is highly imbalanced. So, we need to balance it as well.

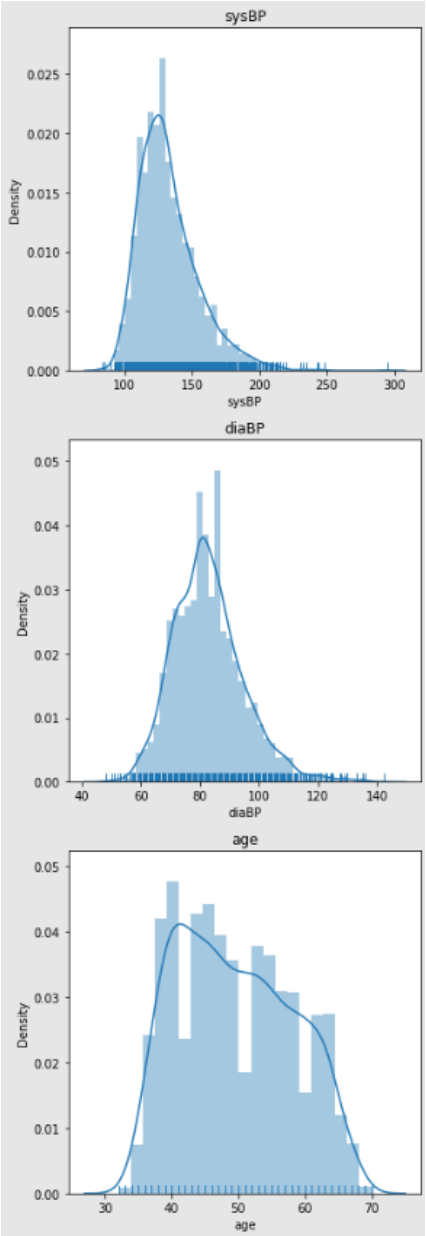


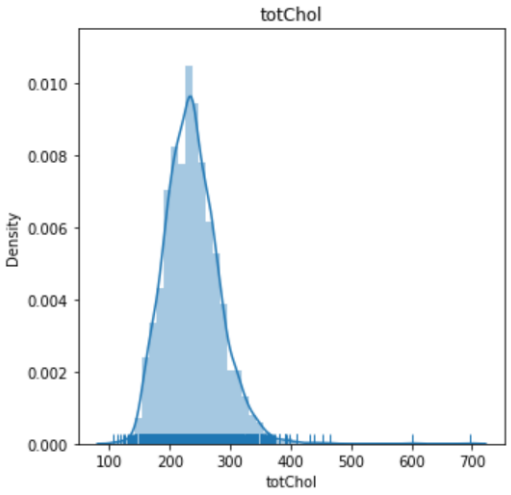
Furthermore, we remove BPMeds, diabetes, and prevalent stroke because of their heavy imbalance.

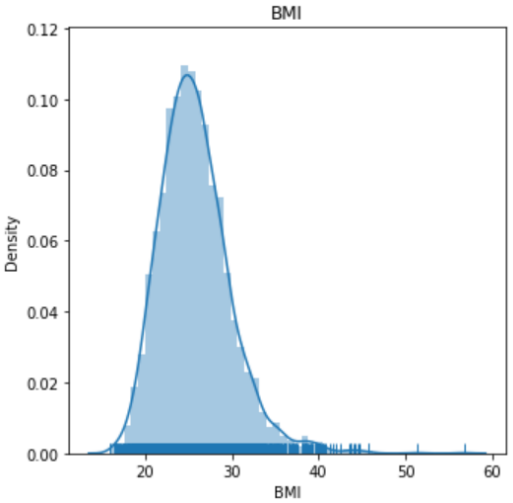
The distributions of continuous variables are as follows:









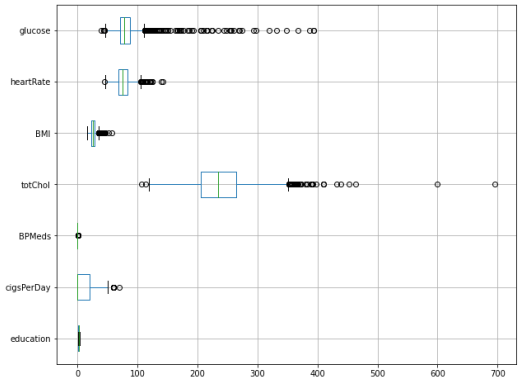


From here we can see that many of the variables are right-skewed. This may be because of outliers, which we will see later in outlier treatment.

4. Null Value treatment

Null values are present in education, CigsPerDay, BPMeds, totChol, BMI, heartrate, and Glucose.

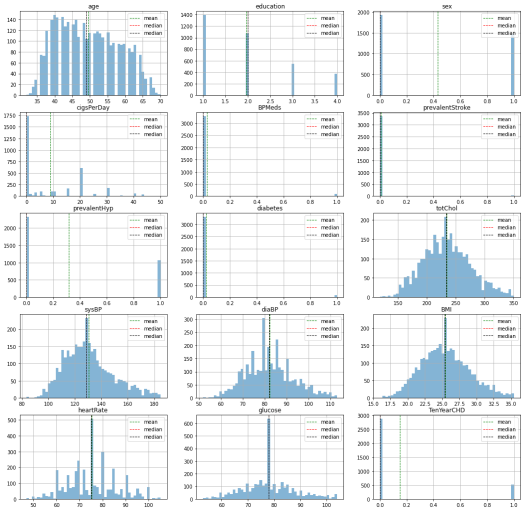
Our options to impute are mean, mode and median. To decide which one to impute we first see the boxplot of the variables.



Since there are outliers in the variables we should not input the mean as the mean is heavily affected by outliers.

For all the variables we see that mode and median are equal. So, we can use either mode or median as these are not affected by outliers.

Hence, there are no more null values in the dataset.



5. Outlier Treatment:

As seen before many of the forementioned variables have outliers. Some of the values are absurd for example the blood glucose levels, in which we have values well above 400.

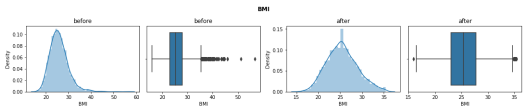
So, our aim now is to decide what should be the value with which we should replace the outliers with.

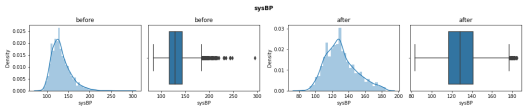
First, we define the upper and lower i.e.,

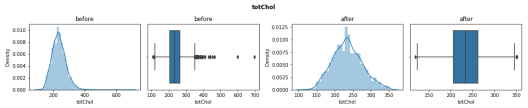
$q_1 - 1.5 * (q_3 - q_1)$ *and*

$q_3 + 1.5 * (q_3 - q_1)$ respectively.

All the values below the lower limit and above the upperlimit are substituted wth the median of the variables,

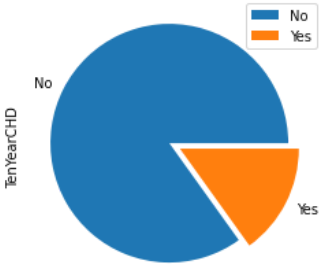






6. Handling Class Imbalance:

We see that the target variable i.e., 'TenYearCHD' is heavily imbalanced.



So, we have to fix it and in order to do that we use the following:

1. **SMOTE** (Synthetic Minority Oversampling Technique)

2. Tomeklinks

3. Penalizing misclassification.

Since we have a small dataset we try to get more data points using SMOTE then cut a few down using Tomek and then for some(RandomForest, SVC) of the classifier models we penalize the loss function.

7. Fixing Non-Numeric Variables:

In our model, we have only two variables containing non-numeric values sex and is_smoking.

We drop is_smoking because its information is obtained from the number of cigs per day column.

Next, we replace Female with 0 and Male with 1.

8. Preparing Data for logistic regression.

To make sure our data has no multicollinearity we make a column called glucose_age_chol which is a function of glucose, age, and chol. Where age and chol contribute the most.

In the end the VIF values for the variables are as follows:

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

9. Train Test split and oversampling for logistic regression

For logistic regression, we need to remove multicollinearity which is done. Next, we need to scale the data by MinMaxScaler.

i.e., for each data point at each variable, we replace each value with

$$\frac{datapoint - minimum}{maximum - minimum}$$

Then we split the data into train and test set, 80% of data goes to the train set and rest to test set. The test set contains 102 values as 1. The train set contains 2303 values as 0 and 409 values as 1.

Now, we have to balance the 1's in the training dataset. After using SMOTE and Tomeklinks our training set has 1905 values of zeros and ones.

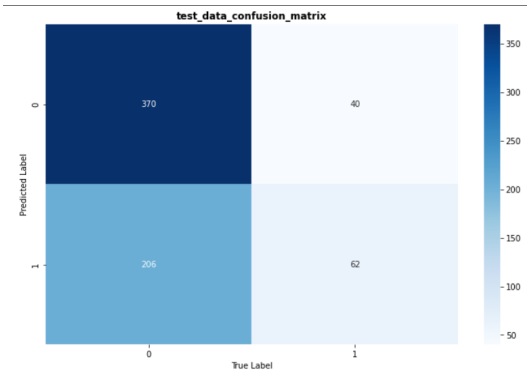
This same dataset is also used for Naive Bayes model.

10. Applying Classification Model.

Before applying any model we need to decide our aim. Since, our aim mostly should be to predict whether some person is going to have heart diseasein next 10 years or not, we will prioritize predicting the 1*s* and the scoring metric chosen is ROC-AUC curve.

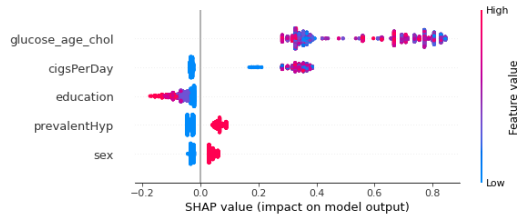
1. Logistic Regression: Using GridSearchCV we get C value as 0.77

and penalty as l2. The confusion matrix formed is as follows.



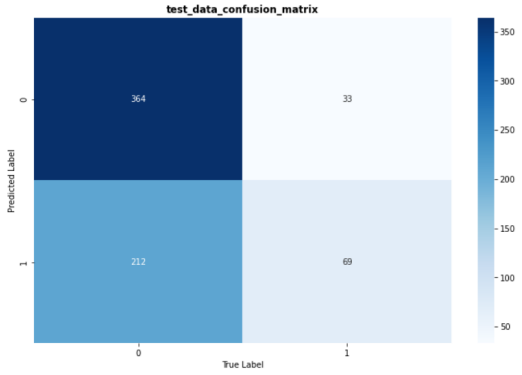
Here we are able to predict 60% of the people who may develop heart disease, our aim is to increase that.

The ROC-AUC curve and Shap values are also obtained.



2. Gaussian Naive Bayes: On applying Naive Bayes model. We canot get better results so we mostly neglect this model.

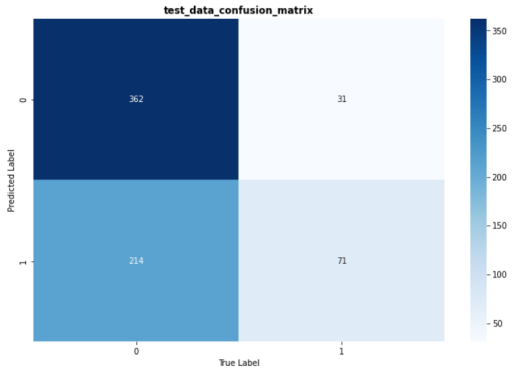
3. Support Vector Classifier 1:Before applying SVC we prepare the dataset. For this we use the dataset before removing multicollinearity. On applying support vector classifier use GridSearchCV and obtain this confusion matrix.



Here we obtained a better result than Logistic Regression. Same as before we obtain the ROC-AUC curve and classification report as well.

4. Support Vector classifier 2: We use two methods to handle the class imbalance, we use smote and tomek links to generate new data points and then we use loss function where misclassifications are heavily penalized.

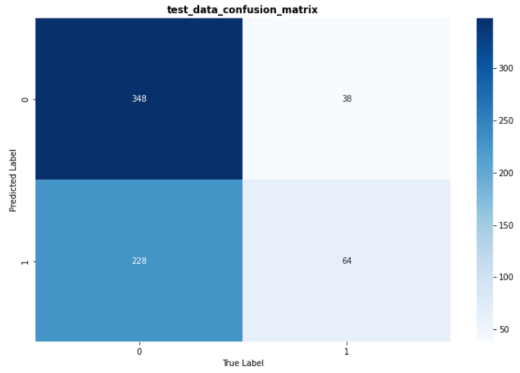
Then we apply SVC with GridSearchCV to get C = 40, gamma = 0.01 and kernel = rbf to get



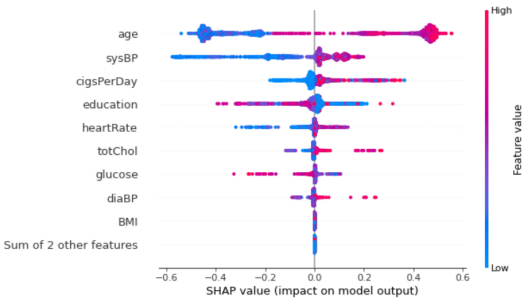
5. Ensemble Tree Models: We prepare the data for ensemble tree models. Since these do not require scaling we create the dataset and train test split accordingly.

5.1. LGBM Implementation: We use GridSearchCV to get the optimum hyperparameters (maximum depth,

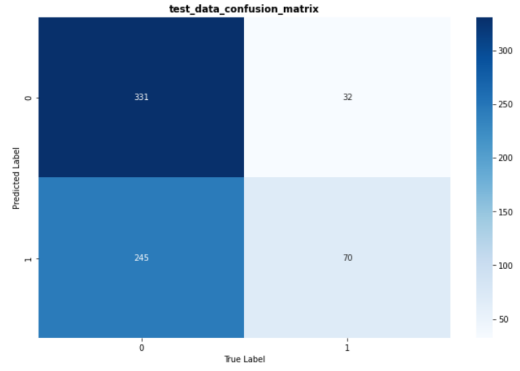
number of estimators and learning rate). The confusion matrix obtained is



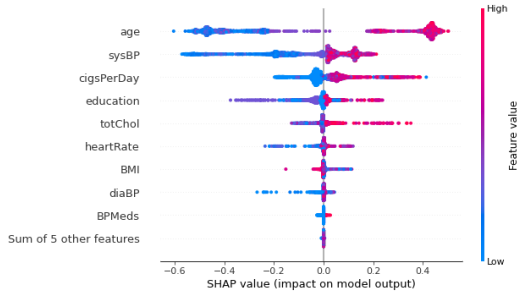
Here we could appoint for a bit more than 60% of the data. The ROC-AUC curve and SHAP plots are also obtained.



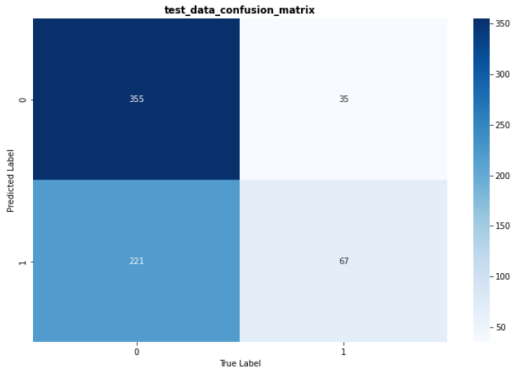
5.2. Extreme Gradiend Random Forest Boosting Model: We implement XGRFB model and tune the hyperparameters using GridSearchCV. The confusion matrix hence obtained is



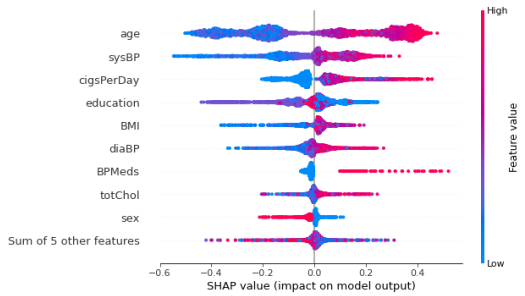
Further we obtain the ROC-AUC curves and SHAP values as well.



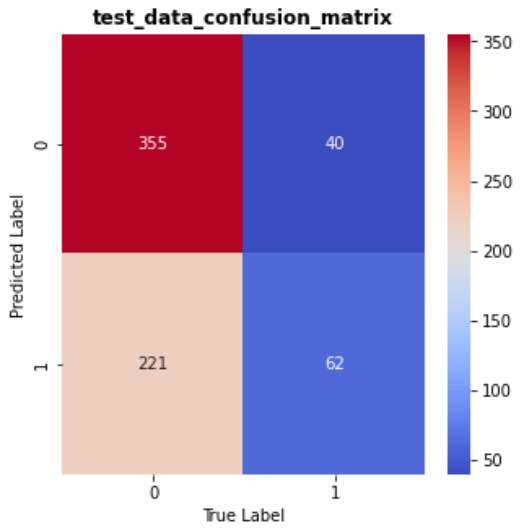
5.3. Random Forest Implementation: Same as before we apply Random Forest Model and use GridSearchCV to tune the hyperparameters. The confusion matrix obtained is



Further, we plot ROC-AUC curve and plot the SHAP values.



6.Neural Network: We apply Neural Netwoek model with two hidden layers and one output vertex with early stopping. The confusion matrix obtained is



At last we plot the shap values as well.

10.Conclusion.

The confusion matrix, classification report and many other metrics are obtained and the best values are chosen.