Capstone Project

Cardio Vascular Risk Analysis

Team Members

Mujahid Sayyed
Sarthak Gupta
Lova Kumar Poluparti

Contents

- Introduction
- Problem Statement
- Methodology

    (1) Loading the data

    (2) Exploratory Data Analysis

    (3) Treating missing values and outliers

    (4) Feature engineering

    (5) Train test split

    (6) Data Modeling

    (7) Cross validation

    (8) Hyper parameter tuning

- Conclusion

Introduction

- Cardiovascular disease is a major health burden worldwide in the 21st century.
- Consequently, there is an urgent requirement for early location and treatment of such diseases.
- Machine Learning is one of the slanting innovations utilized in numerous circles far and wide including the medicinal services application for predicting illnesses.
- In this project, data of Framingham city is given.
- The Framingham Heart Study is a long-term, ongoing cardiovascular cohort study of residents of the city of Framingham, Massachusetts.

Problem Statement

- The dataset is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts.
- The classification 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.

Numerical columns:

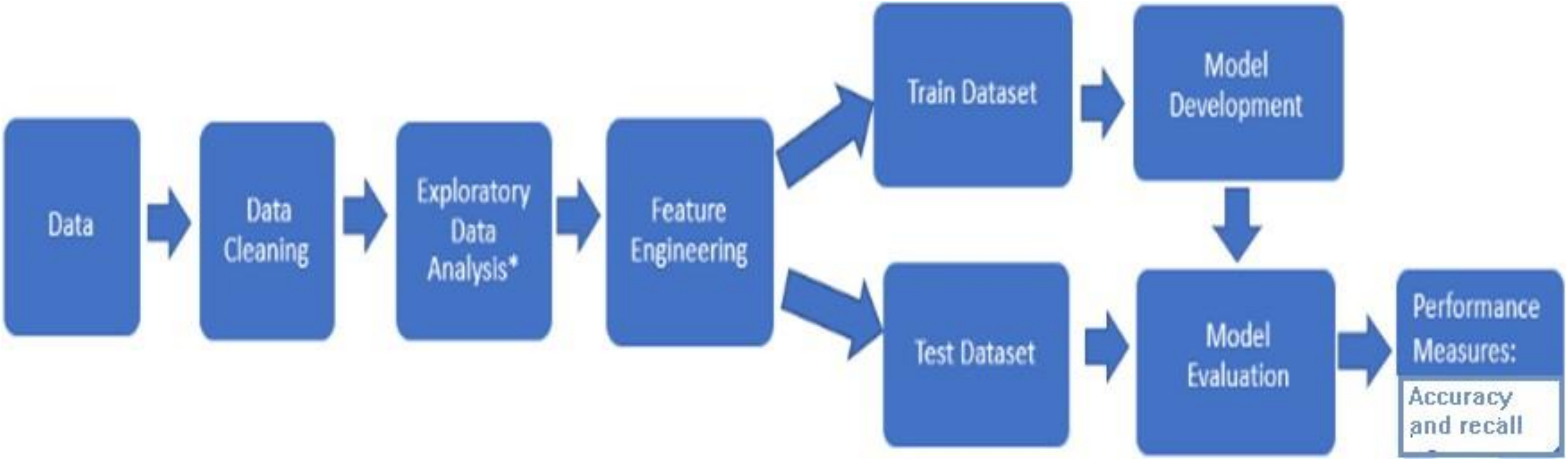
id, age, totChol, sysBP, diaBP, BMI, heartRate, glucose

Categorical columns:

education, cigsPerDay, sex, is_smoking, BPMeds, prevalentStroke, prevalentHyp, diabetes, TenYearCHD

Dataset shape: (3390, 17)

Methodology



Data

Data Cleaning

Exploratory Data Analysis*

Feature Engineering

Train Dataset

Test Dataset

Model Development

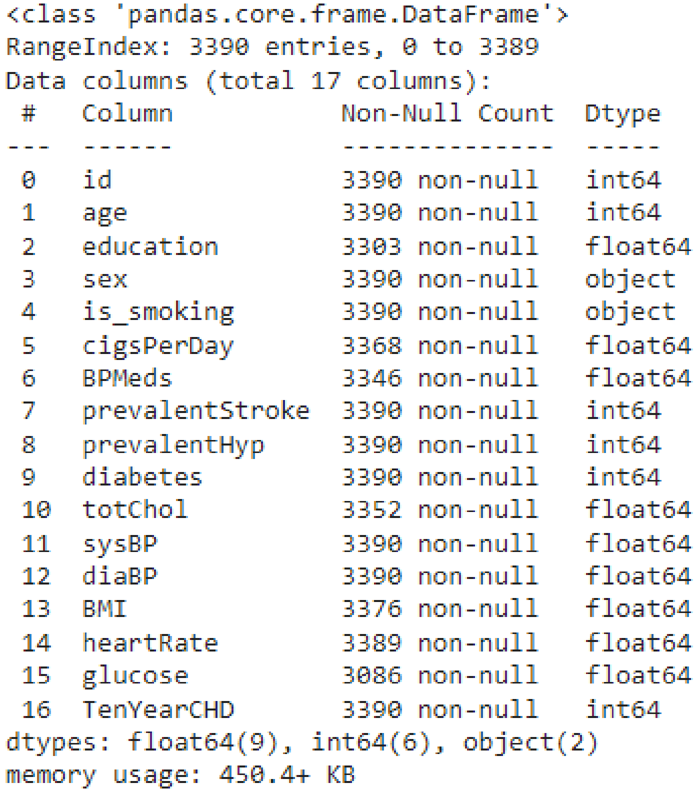
Model Evaluation

Performance Measures:

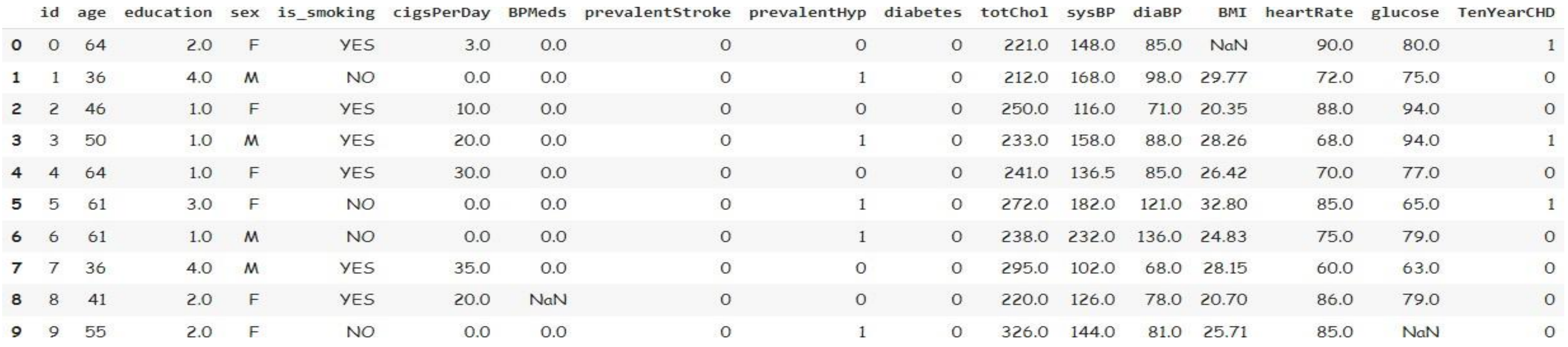
Accuracy and recall

Loading the Dataset

- After loading the dataset we can observe that it contains 3390 rows and 17 columns to deal with.
- Since we have information of all the columns, we can proceed EDA

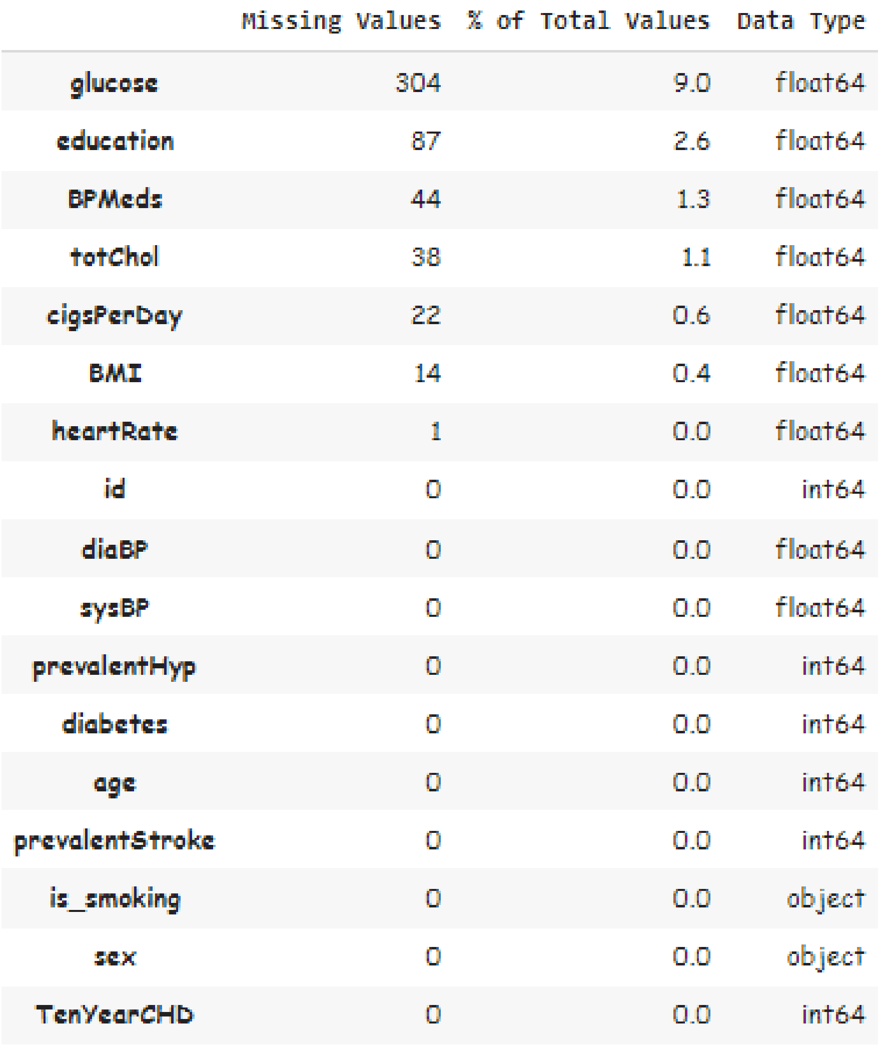


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3390 entries, 0 to 3389
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               3390 non-null   int64  
 1   age              3390 non-null   int64  
 2   education        3303 non-null   float64
 3   sex              3390 non-null   object 
 4   is_smoking       3390 non-null   object 
 5   cigsPerDay       3368 non-null   float64
 6   BPMeds           3346 non-null   float64
 7   prevalentStroke  3390 non-null   int64  
 8   prevalentHyp     3390 non-null   int64  
 9   diabetes         3390 non-null   int64  
 10  totChol          3352 non-null   float64
 11  sysBP            3390 non-null   float64
 12  diaBP            3390 non-null   float64
 13  BMI              3376 non-null   float64
 14  heartRate        3389 non-null   float64
 15  glucose          3086 non-null   float64
 16  TenYearCHD       3390 non-null   int64  
dtypes: float64(9), int64(6), object(2)
memory usage: 450.4+ KB
```



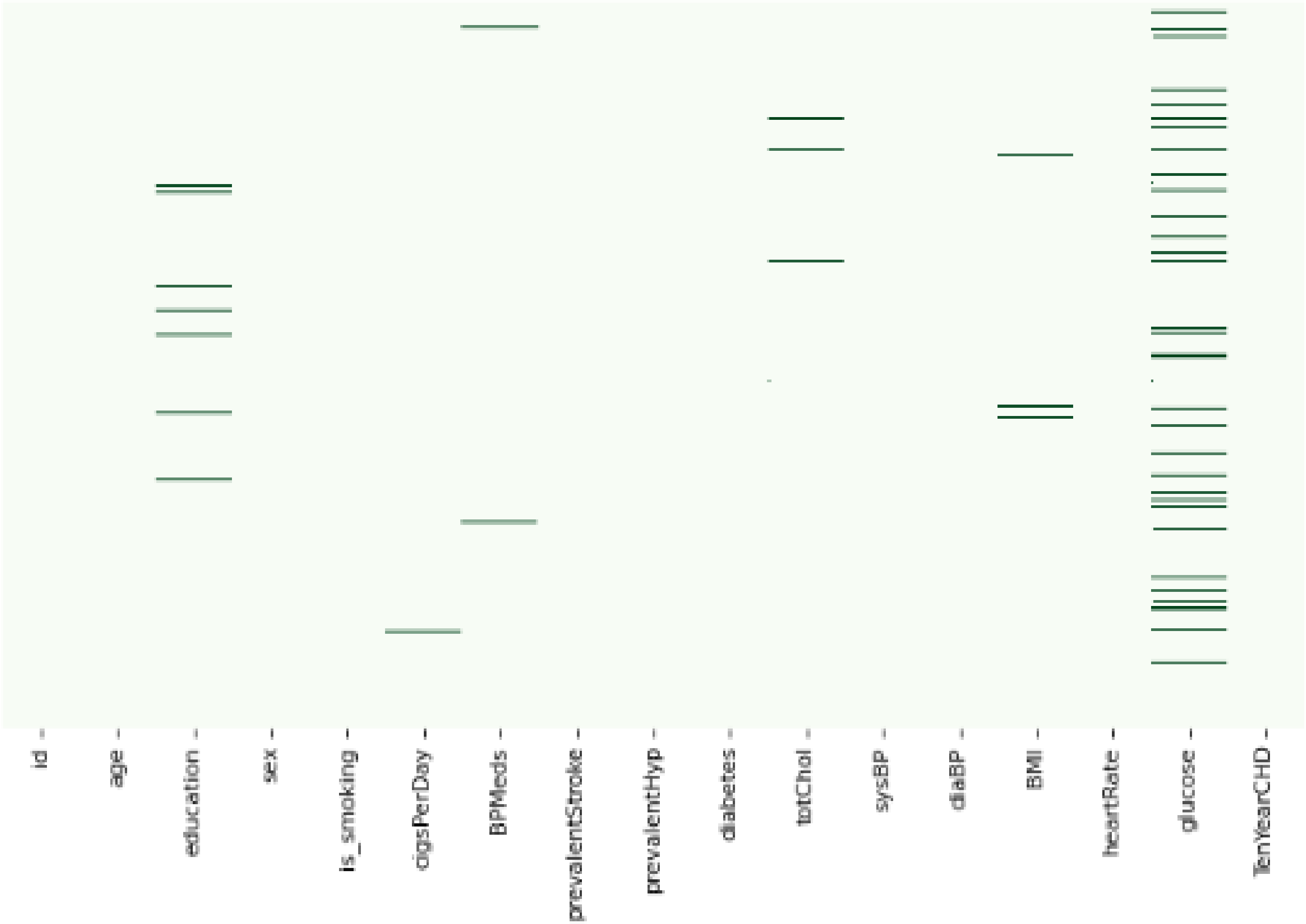
	id	age	education	sex	is_smoking	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
0	0	64	2.0	F	YES	3.0	0.0	0	0	0	221.0	148.0	85.0	NaN	90.0	80.0	1
1	1	36	4.0	M	NO	0.0	0.0	0	1	0	212.0	168.0	98.0	29.77	72.0	75.0	0
2	2	46	1.0	F	YES	10.0	0.0	0	0	0	250.0	116.0	71.0	20.35	88.0	94.0	0
3	3	50	1.0	M	YES	20.0	0.0	0	1	0	233.0	158.0	88.0	28.26	68.0	94.0	1
4	4	64	1.0	F	YES	30.0	0.0	0	0	0	241.0	136.5	85.0	26.42	70.0	77.0	0
5	5	61	3.0	F	NO	0.0	0.0	0	1	0	272.0	182.0	121.0	32.80	85.0	65.0	1
6	6	61	1.0	M	NO	0.0	0.0	0	1	0	238.0	232.0	136.0	24.83	75.0	79.0	0
7	7	36	4.0	M	YES	35.0	0.0	0	0	0	295.0	102.0	68.0	28.15	60.0	63.0	0
8	8	41	2.0	F	YES	20.0	NaN	0	0	0	220.0	126.0	78.0	20.70	86.0	79.0	0
9	9	55	2.0	F	NO	0.0	0.0	0	1	0	326.0	144.0	81.0	25.71	85.0	NaN	0

Exploratory Data Analysis



	Missing Values	% of Total Values	Data Type
glucose	304	9.0	float64
education	87	2.6	float64
BPMeds	44	1.3	float64
totChol	38	1.1	float64
cigsPerDay	22	0.6	float64
BMI	14	0.4	float64
heartRate	1	0.0	float64
id	0	0.0	int64
diaBP	0	0.0	float64
sysBP	0	0.0	float64
prevalentHyp	0	0.0	int64
diabetes	0	0.0	int64
age	0	0.0	int64
prevalentStroke	0	0.0	int64
is_smoking	0	0.0	object
sex	0	0.0	object
TenYearCHD	0	0.0	int64

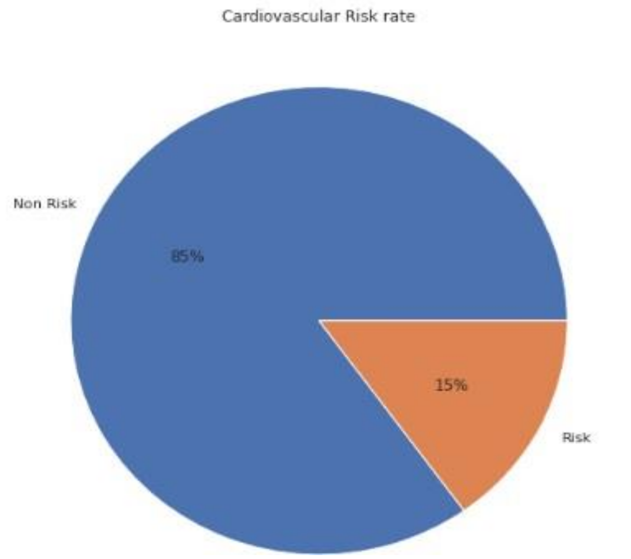
Uni-variate Analysis

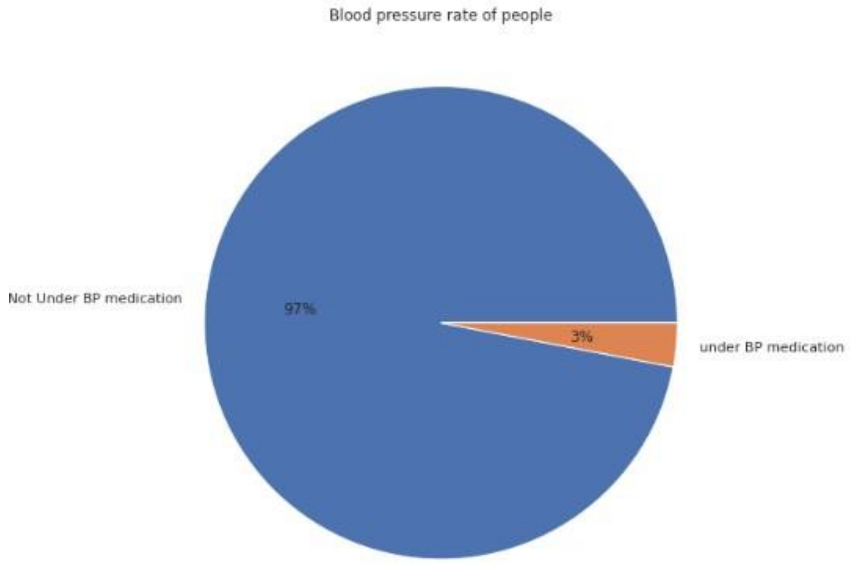


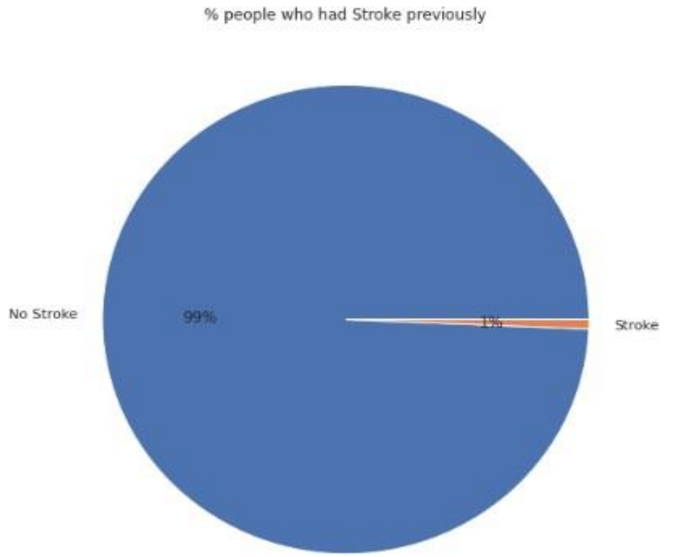
Visualization of Missing values

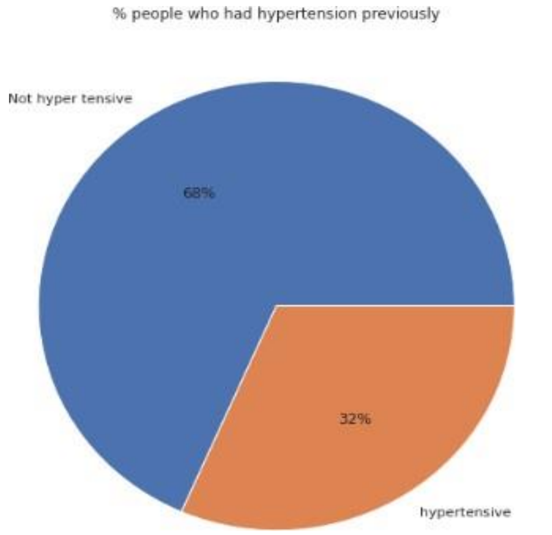
EDA

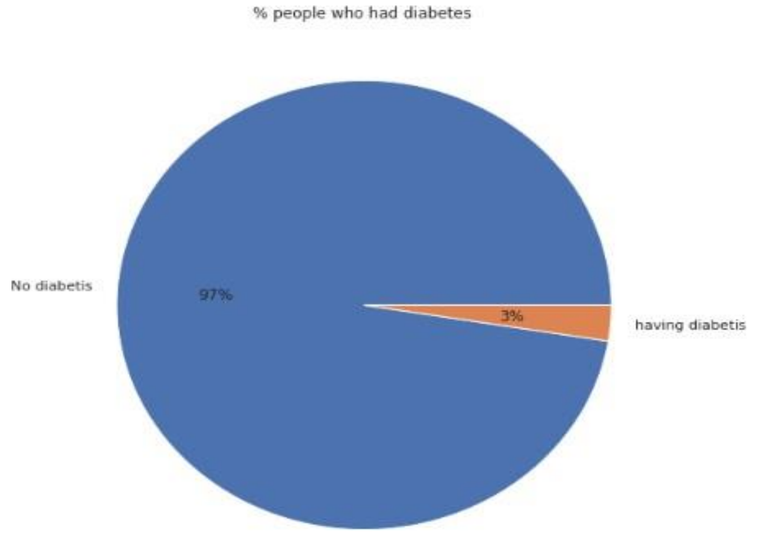
Uni-variate Analysis

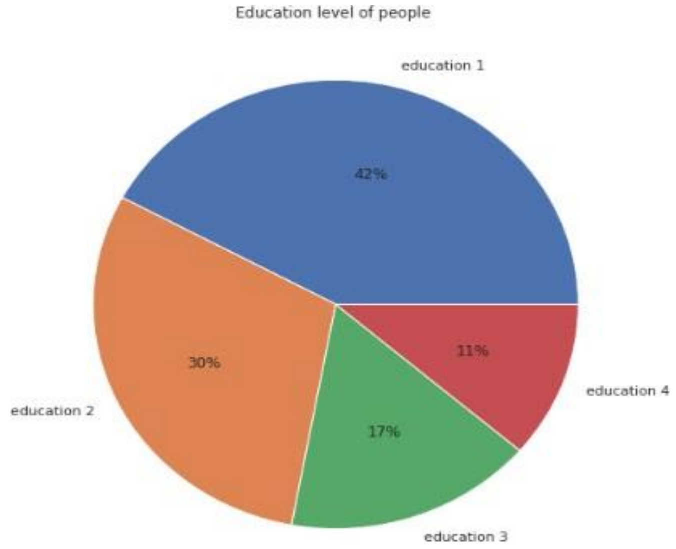

Cardiovascular Risk rate


Blood pressure rate of people


% people who had Stroke previously


% people who had hypertension previously

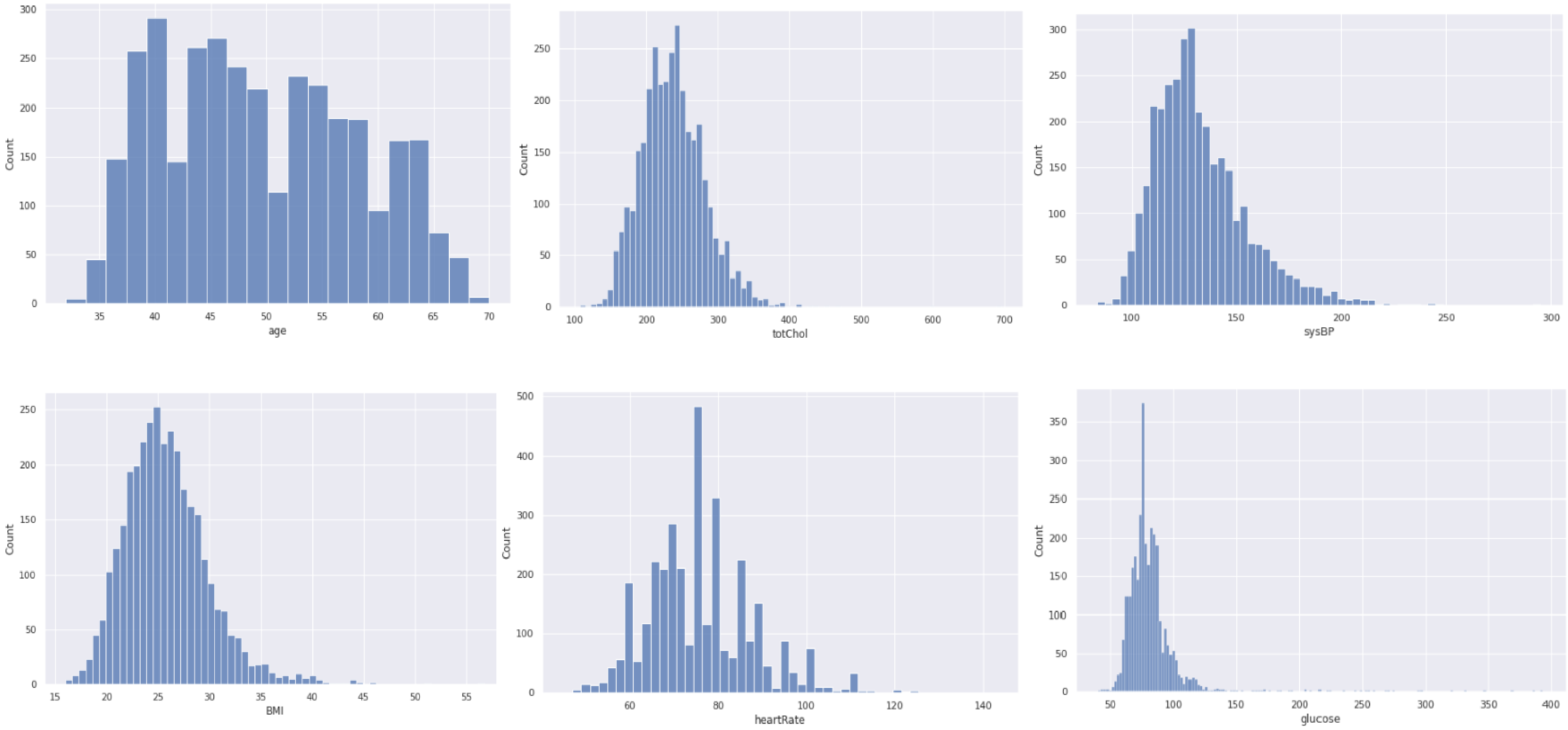

% people who had diabetes


Education level of people

Pie chart to get insight of several columns and it is clear that the data is highly imba

EDA

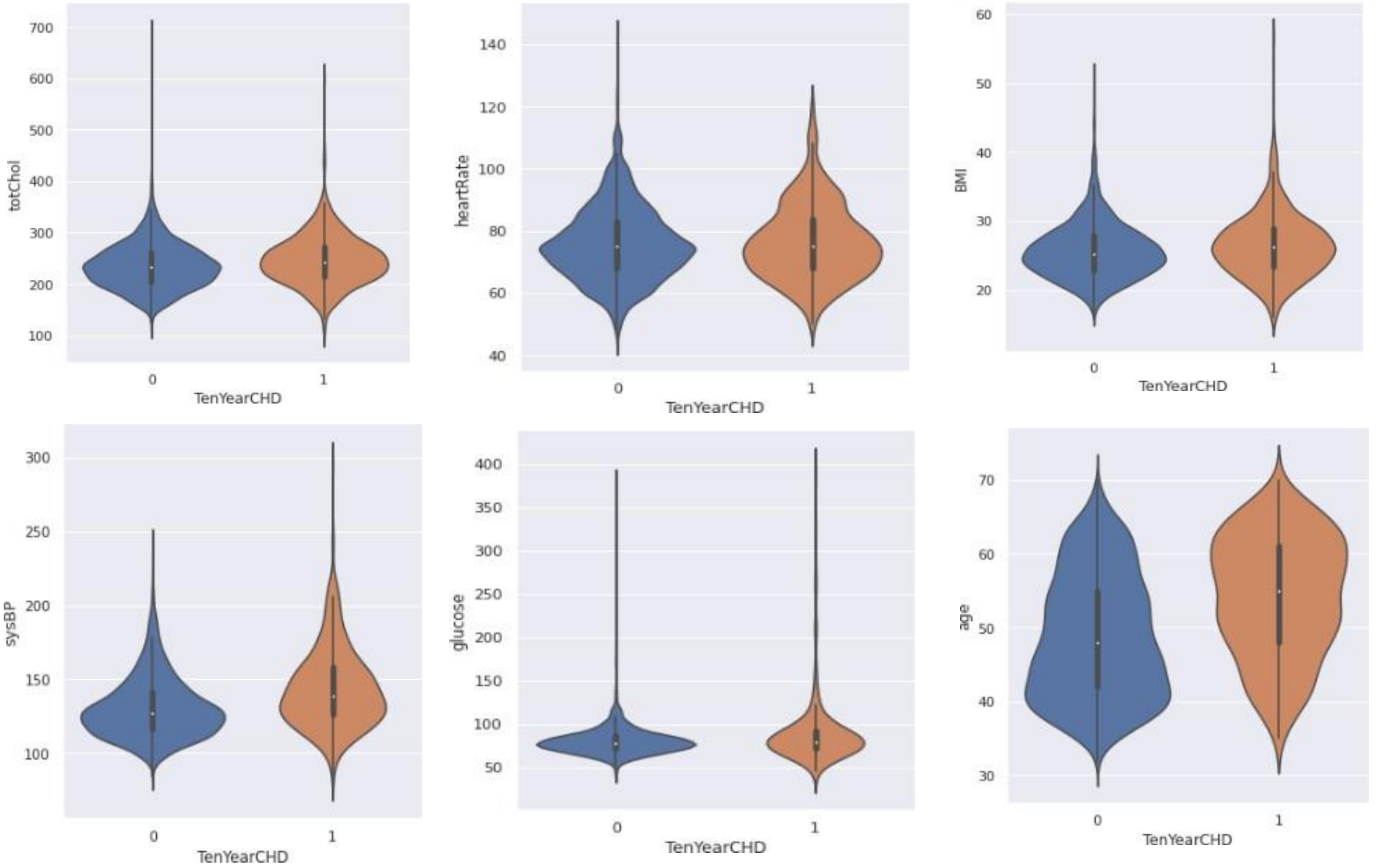
Bi-variate Analysis



Histogram of several numerical columns

EDA

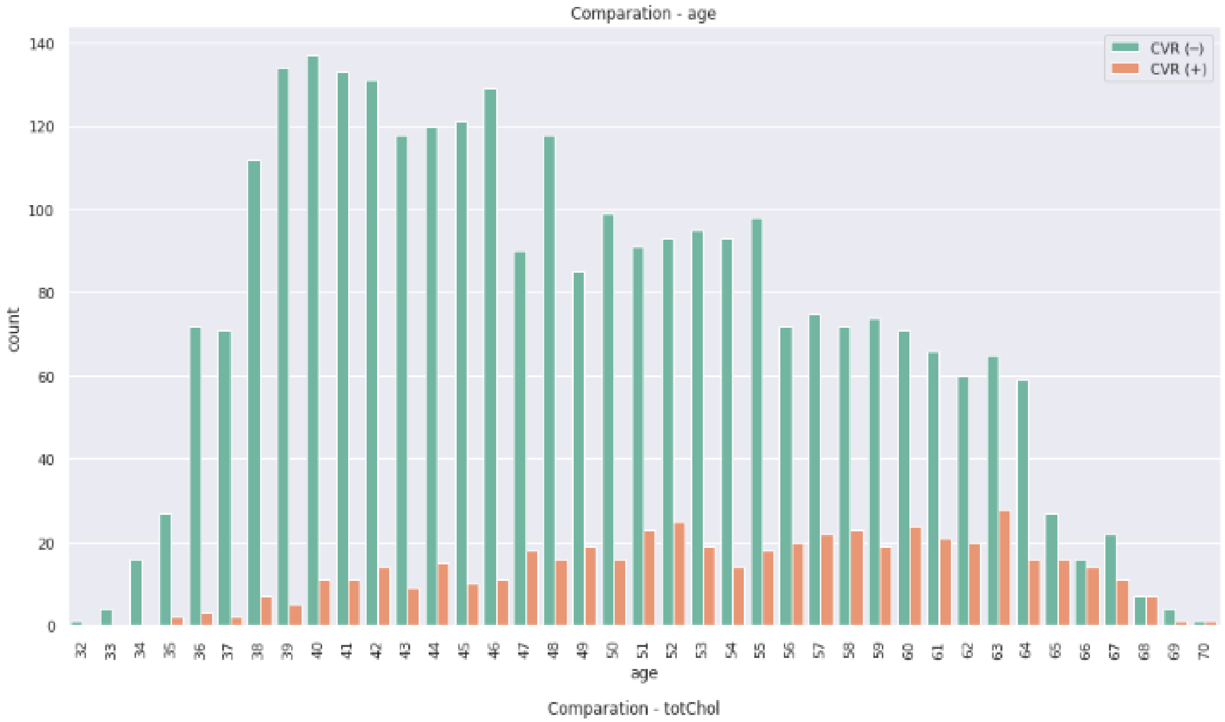
Bi-variate Analysis

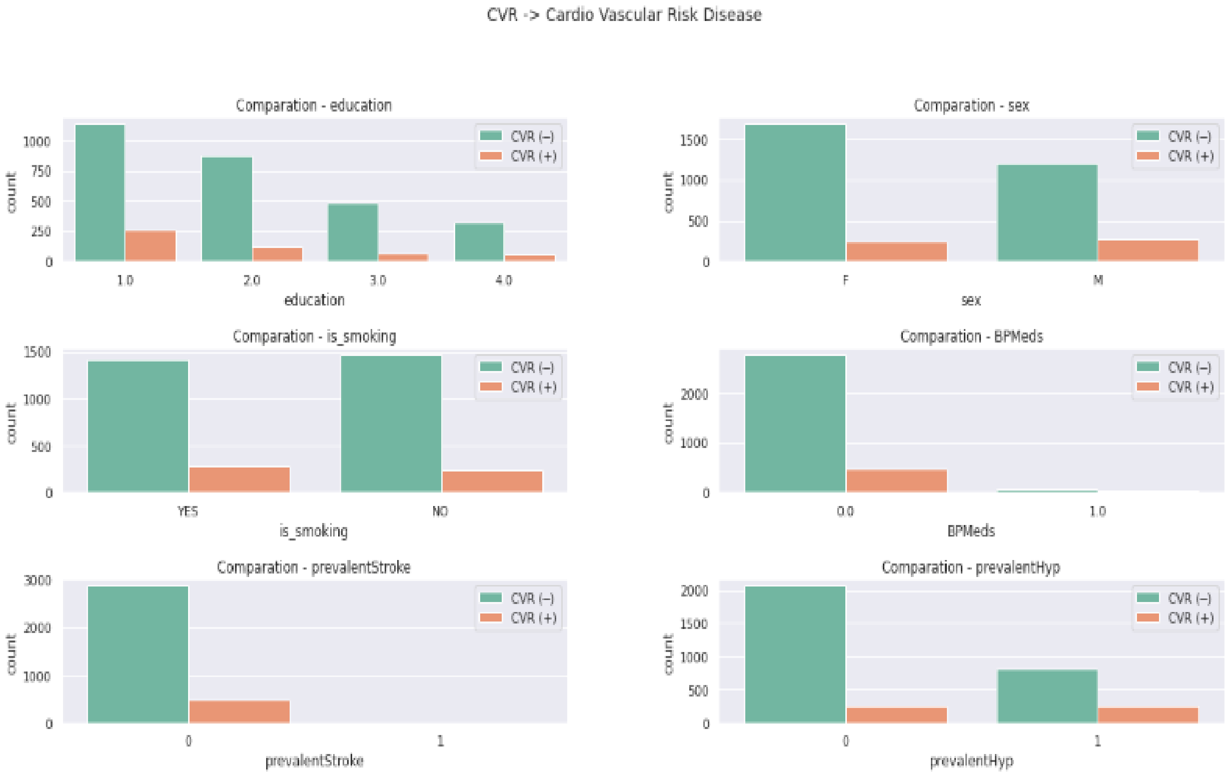


Violin plot for several numerical columns with the dependent variable

EDA

Bi-variate Analysis



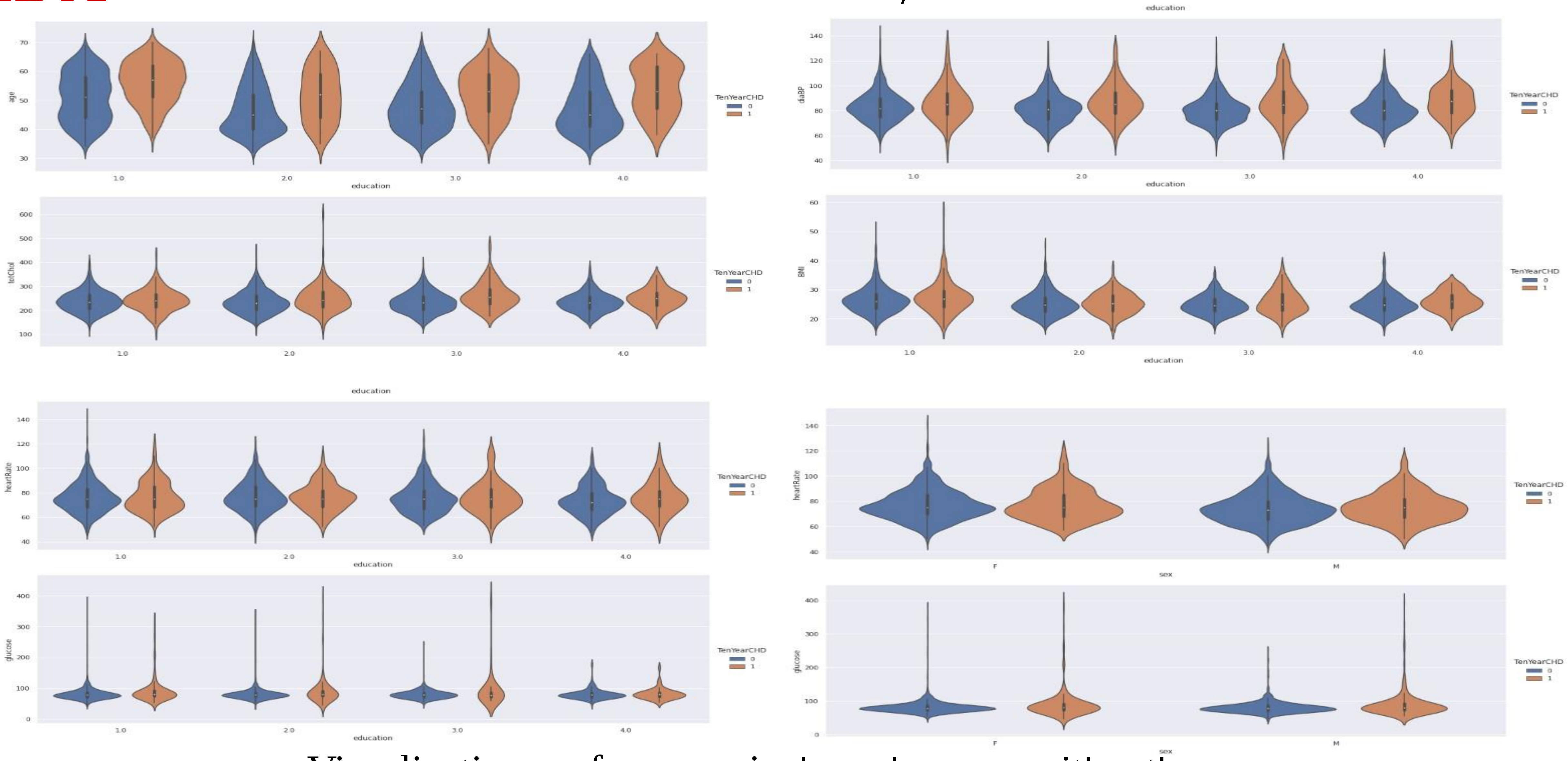


Age column with the dependent variable

Trying to get some insights about the dependent columns with categorical variables

EDA

Multi-variate Analysis

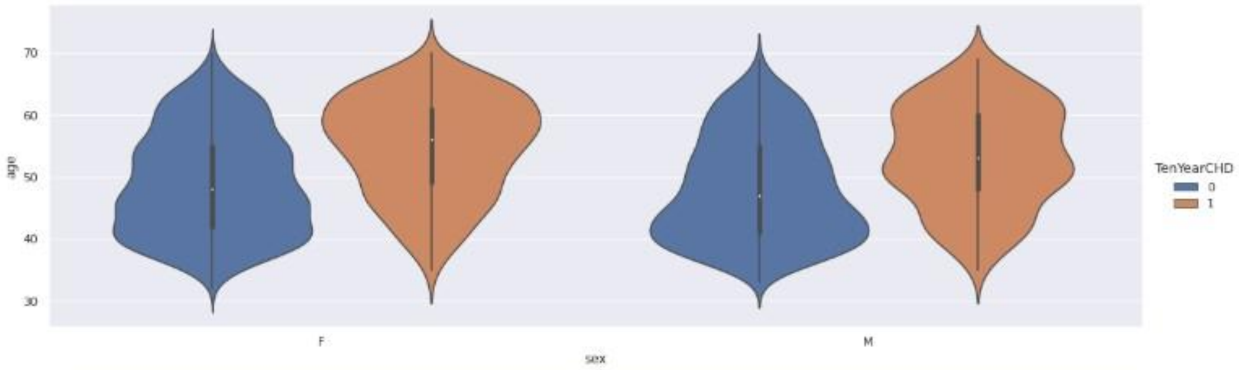


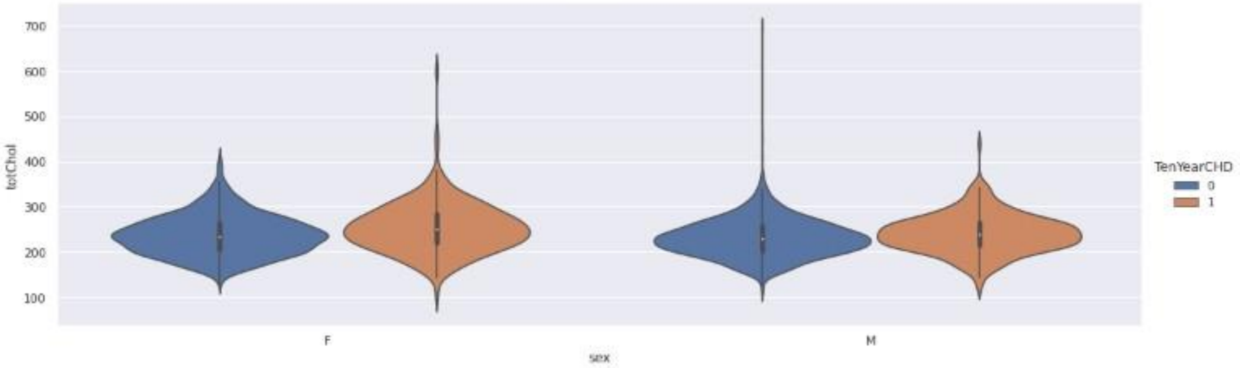
Visualization of numerical columns with the education column along with dependent column

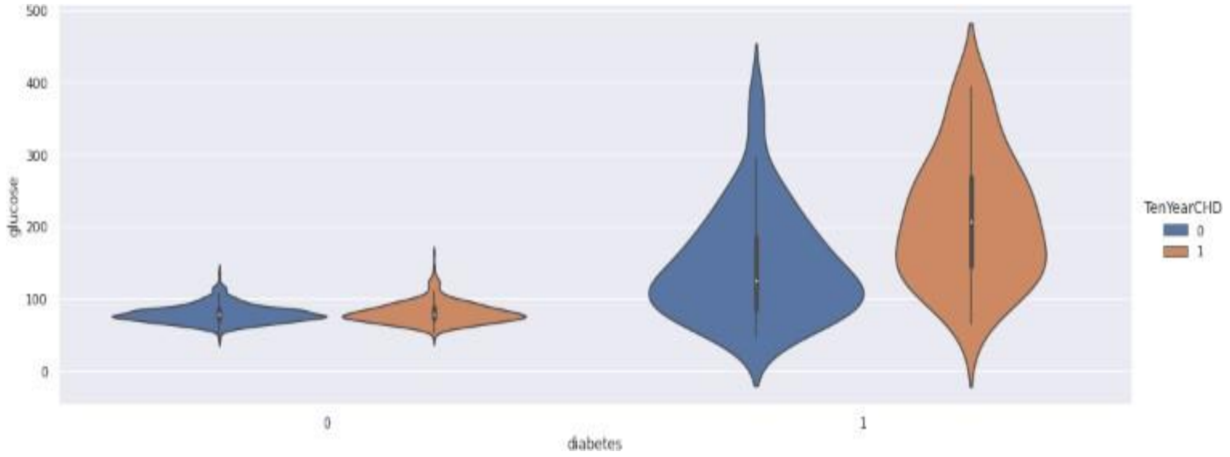
EDA

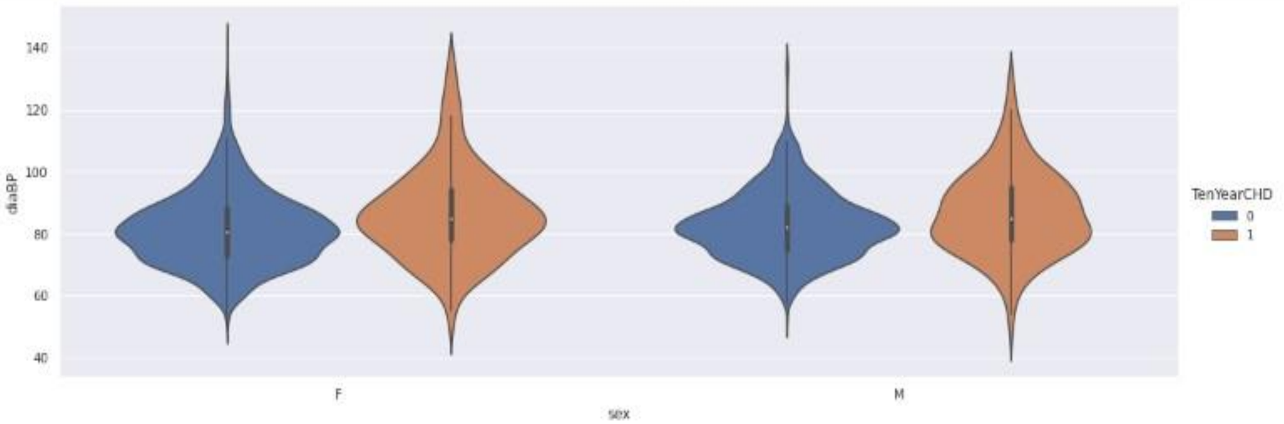
Multi-variate Analysis

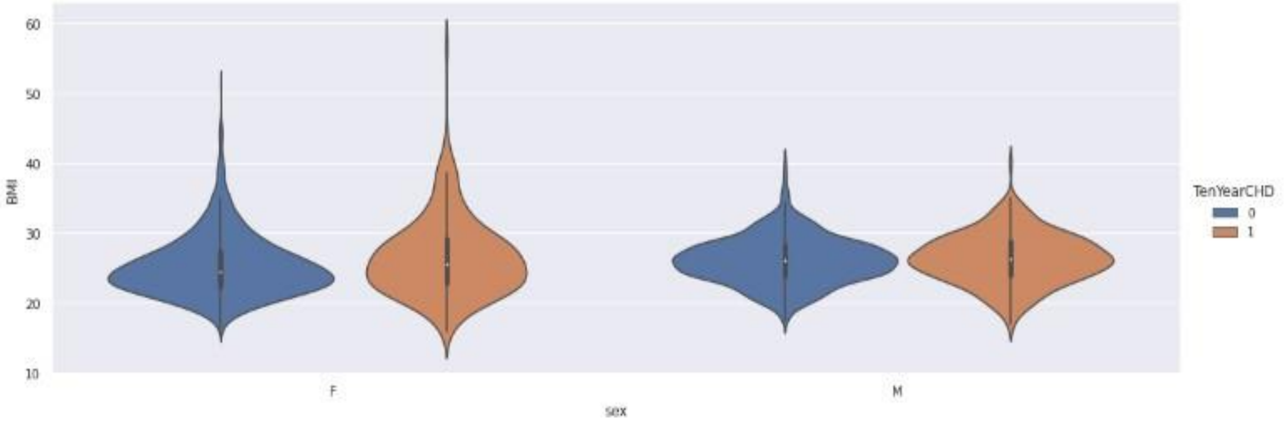
Visualization of numerical columns with the education column along with dependent column



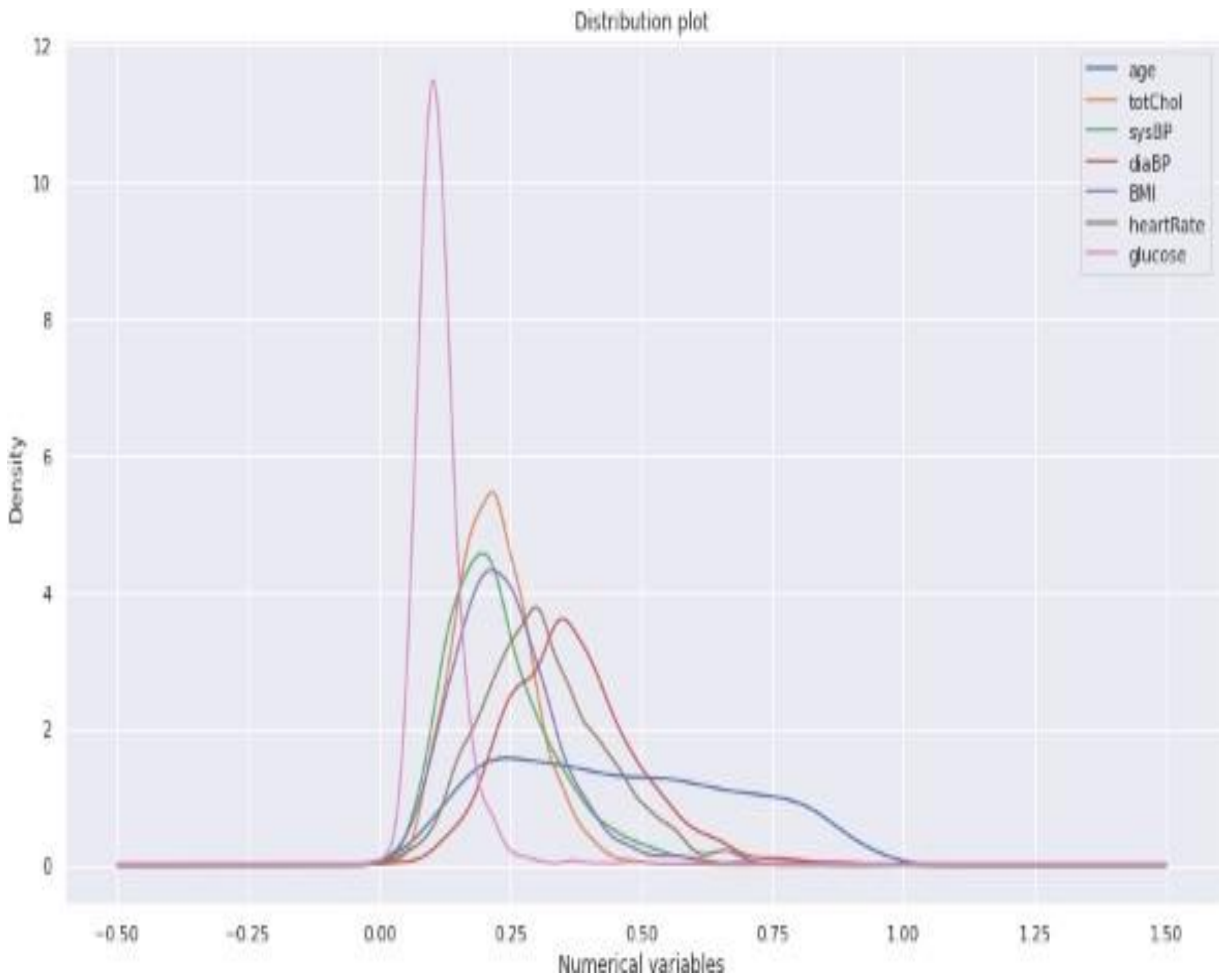








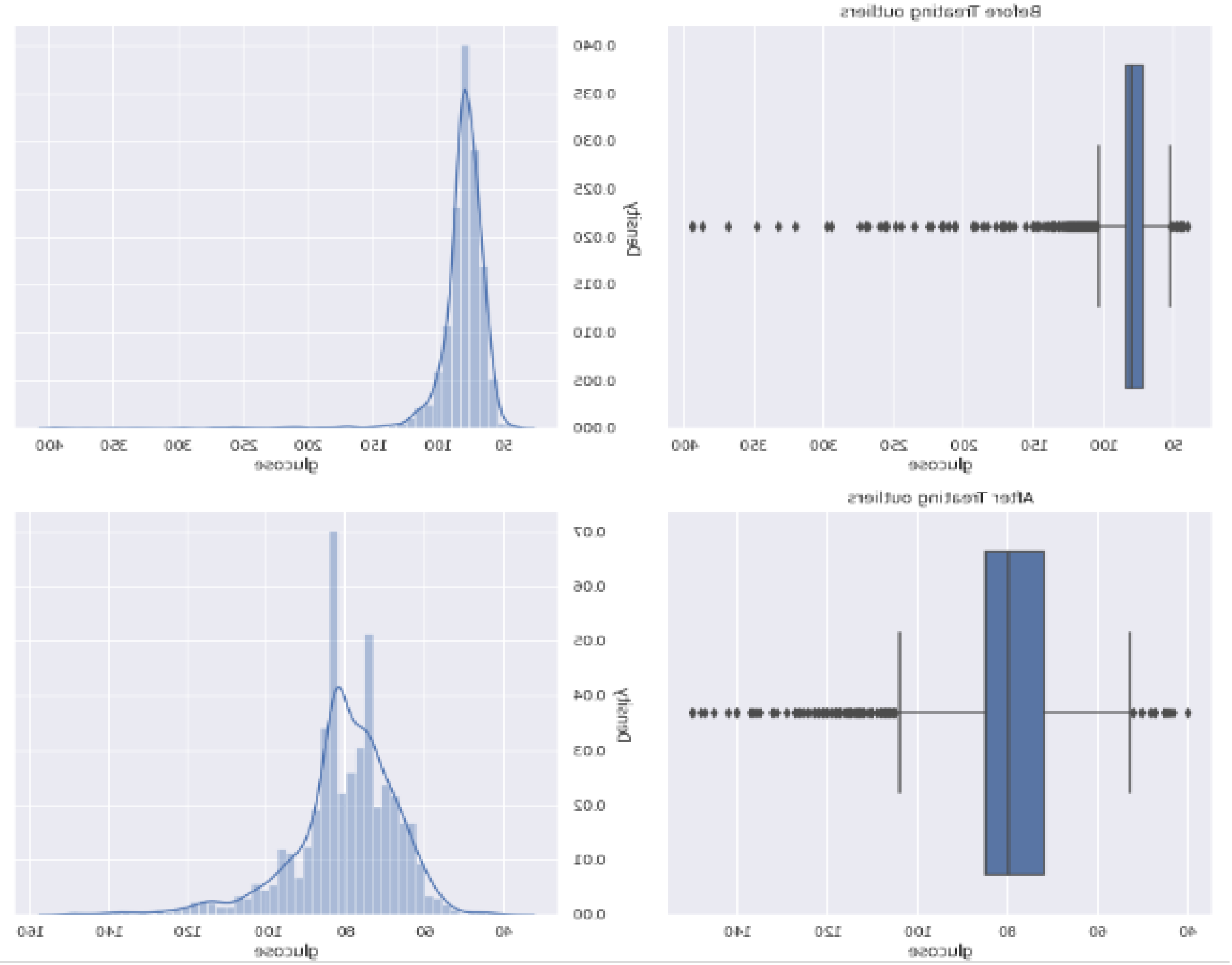
EDA



Conclusions of EDA:

- Slightly more males are suffering from CHD than females.
- The people who has high BMI are at risk of CHD.
- The people with hypertension are at highrisk of CHD.
- The percentage of people who have CHD is almost equal between smokers and non smokers.
- The uneducated people or the people with basic education are at high risk of CHD compared with well educated.

Treatment of Missing Values and Outliers



Treatment of missing values:

- There are 6 columns with missing values.
- All the missing values are treated with mean and mode except the ‘glucose’ column.
- In glucose column, there are 9% of missing values. So we try dealing with this column from KNN imputation.

Treatment of Outliers :

Outliers treatment in this data set is treated by z score.

Feature Engineering

- Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning.

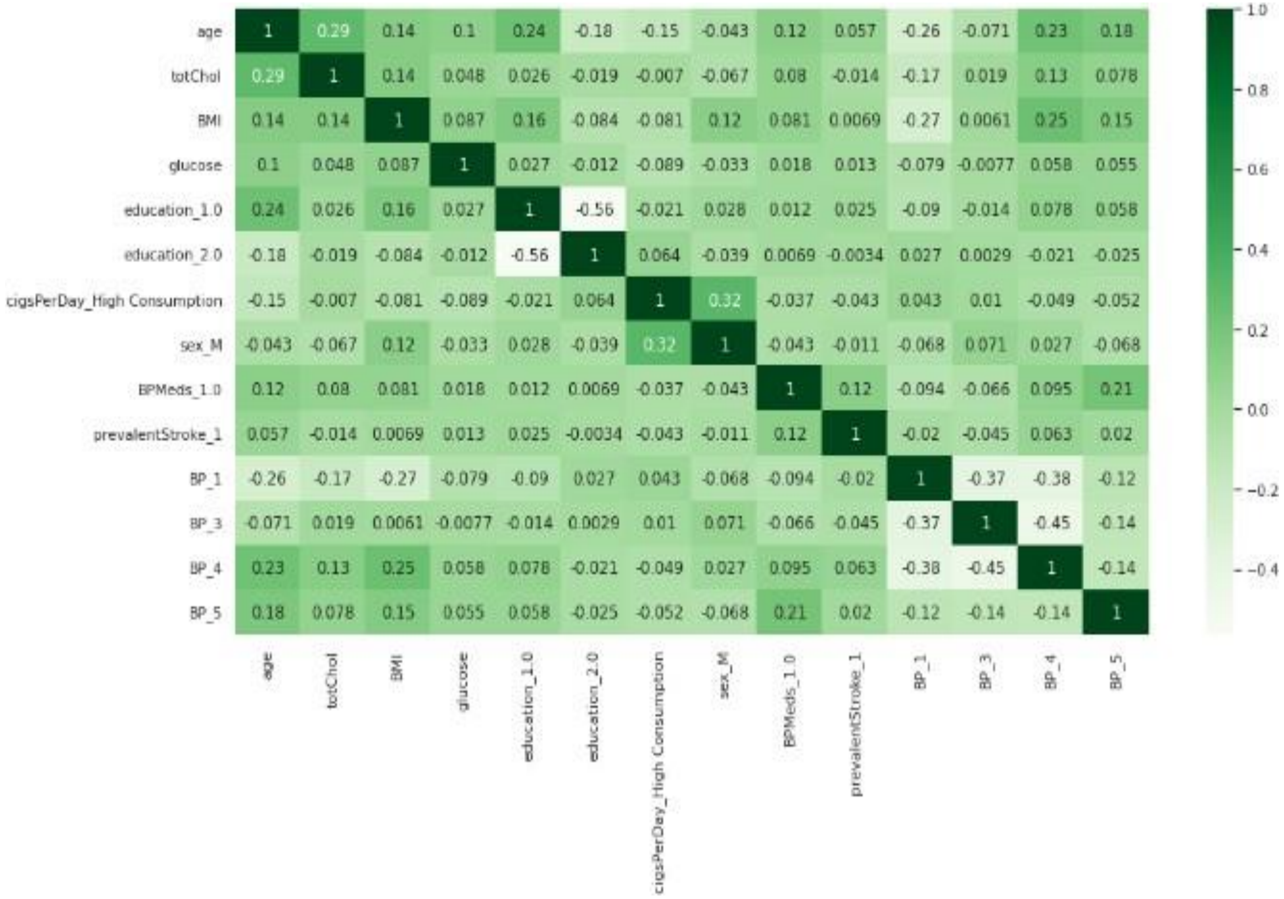
- Feature Engineering consists of various process :
  (1) Feature Creation (2) Transformation (3) Feature Selection

(1) **Feature Creation**: Creating features involves creating new variables which will be most helpful for our model.

(2) **Transformations**: Feature transformation is simply a function that transforms features from one representation to another(Normal distribution). We have used
Box-cox and log transformation to convert columns to Normal distribution.

(1) **Feature Selection**: Feature extraction is the process of extracting features from a data set to identify useful information. We have used f-regression to do the feature selection.

Feature Engineering

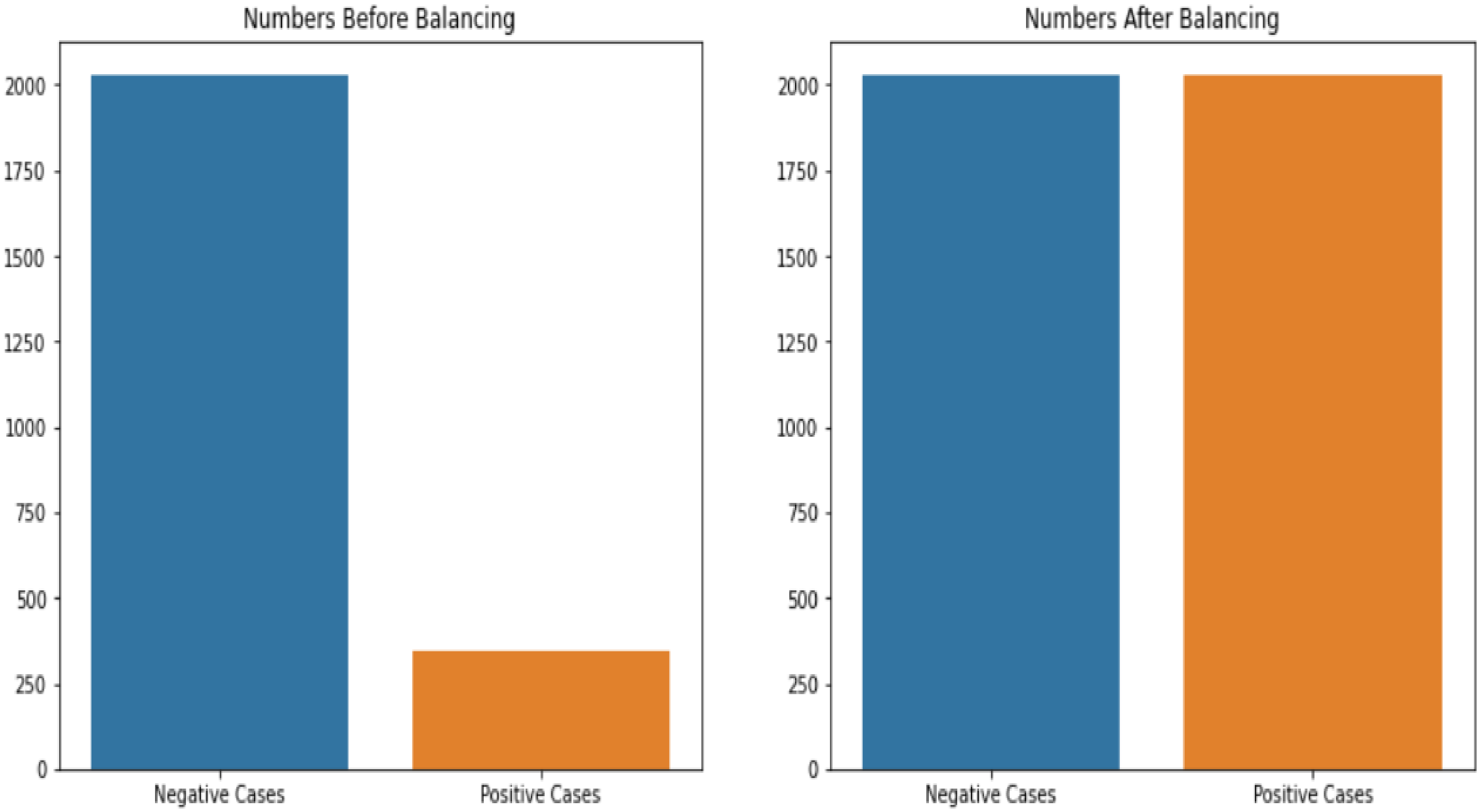


•After the process of feature creation, feature transformation and feature selection, we use the **one hot encoding** to encode all categorical variables.

•Later we carry out the standardization technique of **minmax scaler** to normalize the data.

•Now we are all set with the dataset, so at last we check the **multicollinearity** of the columns. We keep only those columns which are having less or no correlation.

Data Balancing using Smote Model



Shape

trained_dataset shape before smote: 2373

Resampled Tained_dataset shape: 4056

Number of values for class 1&0:

Before smote: {0: 2028, 1: 345}

After smote:{0: 2028, 1: 2028}

Building model

- ✓ Before building the models, we perform the train test split. We have taken 70% of the data as train data and 30% of the data as test data.
- ✓ There are many classification models available in supervised machine learning. The models which we have used are,

  (1) Logistic regression

  (2) Decision Tree

  (3) Random Forest

  (4) K – nearest neighbor

  (5) Naïve Baye's

  (6) Adaboost

  (7) Support Vector Machine (SVM)

Classification Models

1. **Logistic regression** : Logistic regression is a process of modeling the probability of a discrete outcome given an input variable.
2. **Decision Tree** : A decision tree is a mechanical way to make a decision by dividing the inputs into smaller decisions. The tree is divided into decision nodes and leafs. It is based on the concept of **entropy**.
3. **Random Forest** : Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision tree at training time.
4. **K – nearest neighbor** : K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

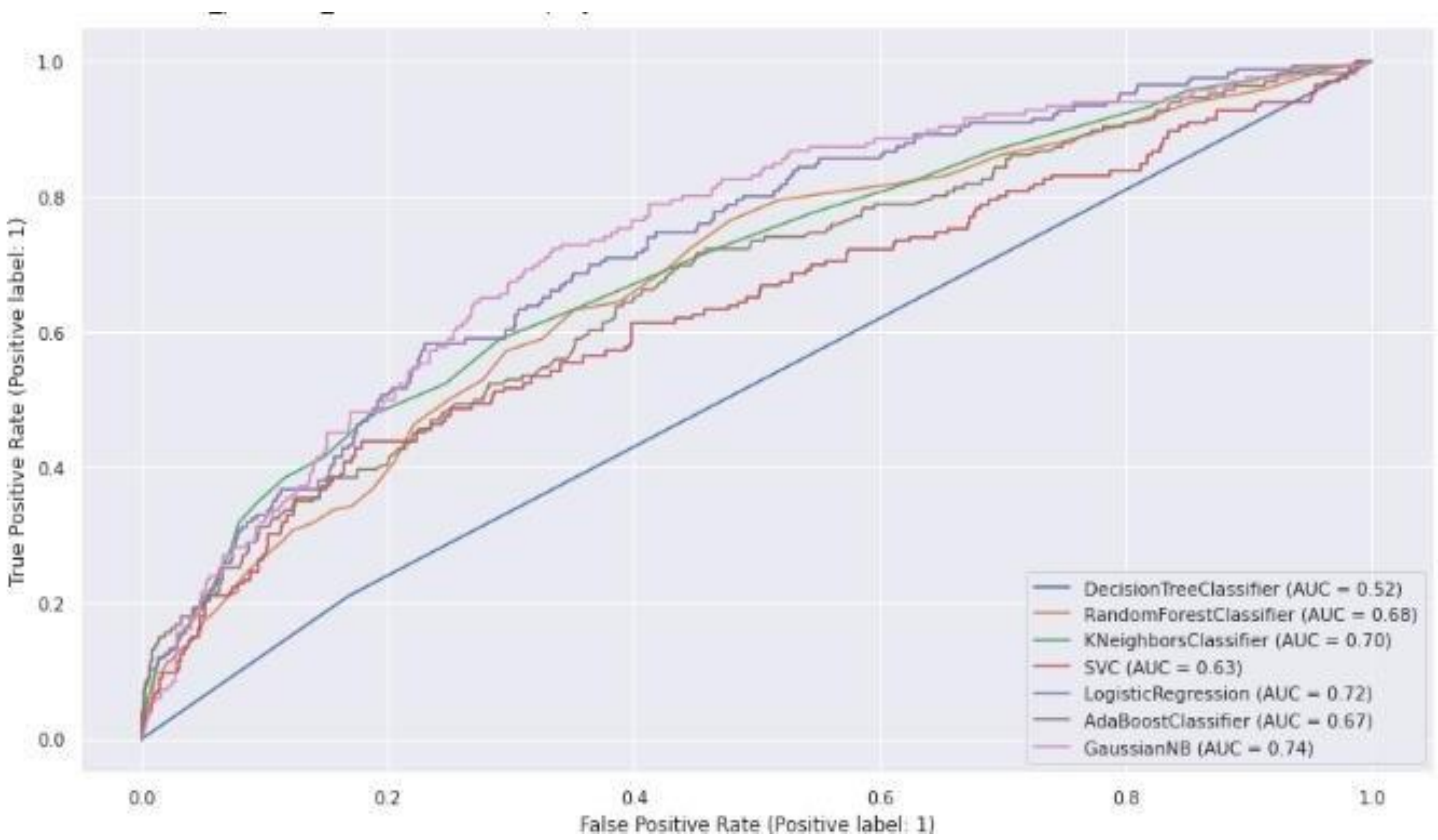
Classification Models

5. Naïve Bayes classifiers : Naïve Bayes classifiers are a family of simple “probabilistic clasifiers” based on applying Bayes theorem with strong independence assumptions between the features.

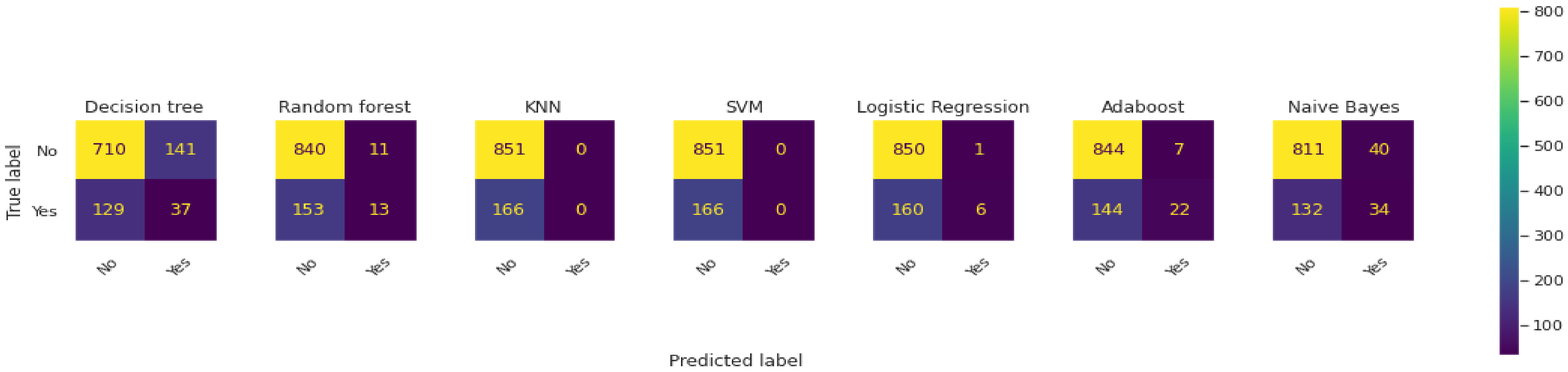
6. Adaboost classifiers : An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

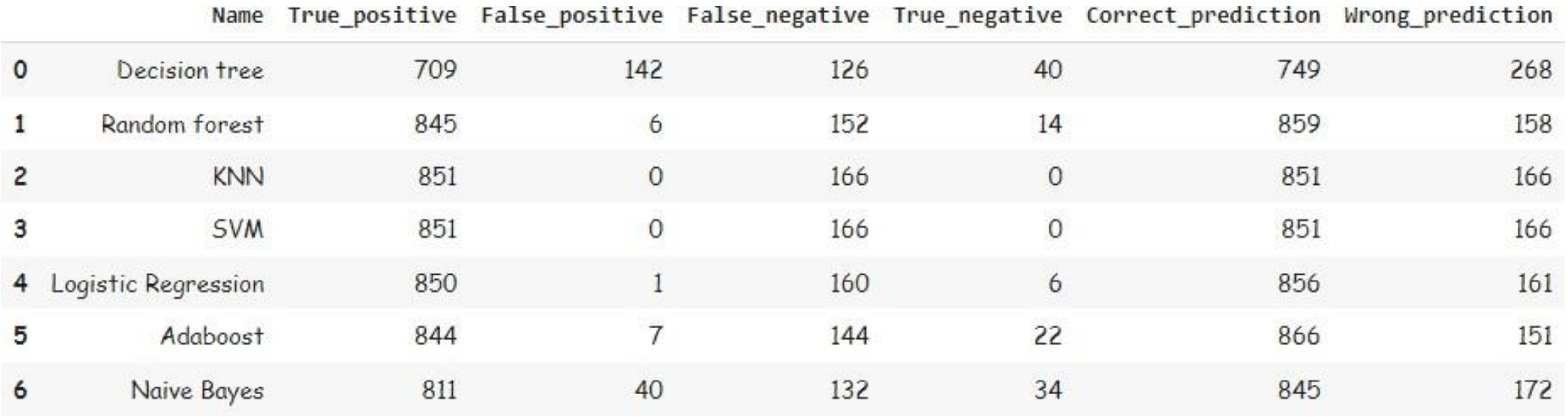
7.Support Vector Machine (SVM) : “Support Vector Machine” (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. Here we plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

ROC of all models



Confusion Metric of all models

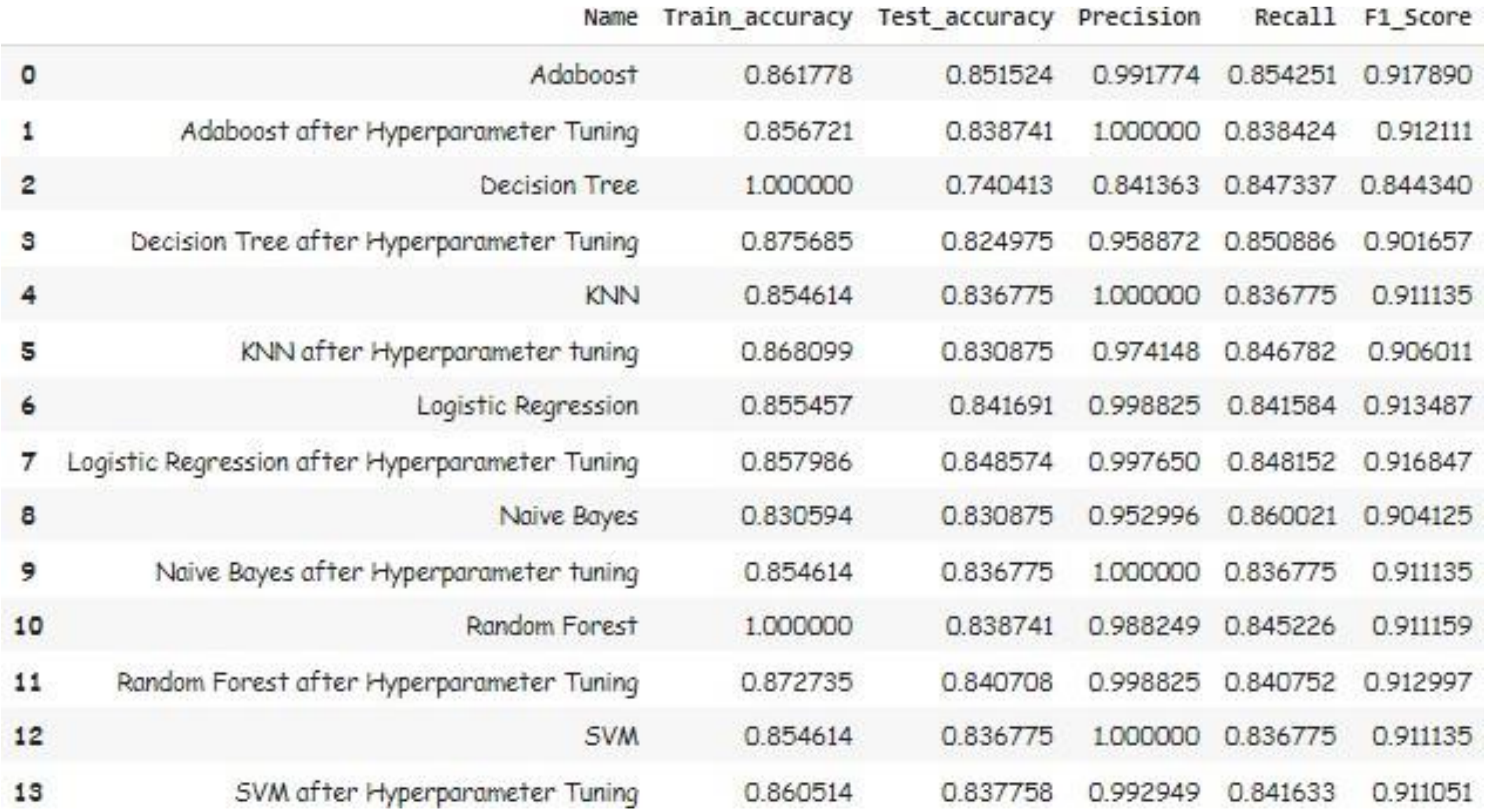




	Name	True_positive	False_positive	False_negative	True_negative	Correct_prediction	Wrong_prediction
0	Decision tree	709	142	126	40	749	268
1	Random forest	845	6	152	14	859	158
2	KNN	851	0	166	0	851	166
3	SVM	851	0	166	0	851	166
4	Logistic Regression	850	1	160	6	856	161
5	Adaboost	844	7	144	22	866	151
6	Naive Bayes	811	40	132	34	845	172

Hyper Parameter Tuning

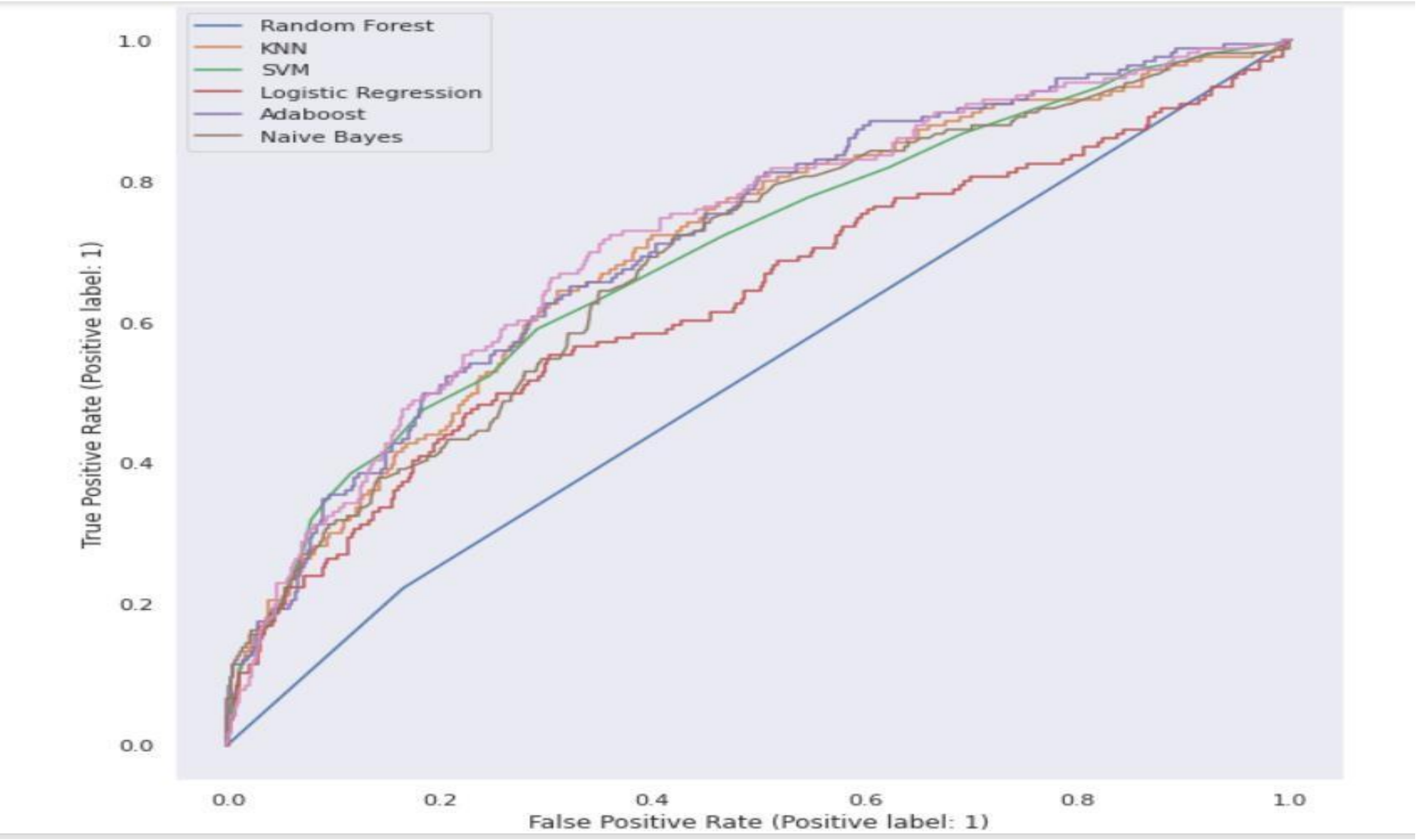
- After building models, we try to improve the performance of the models by hyper parameter tuning.



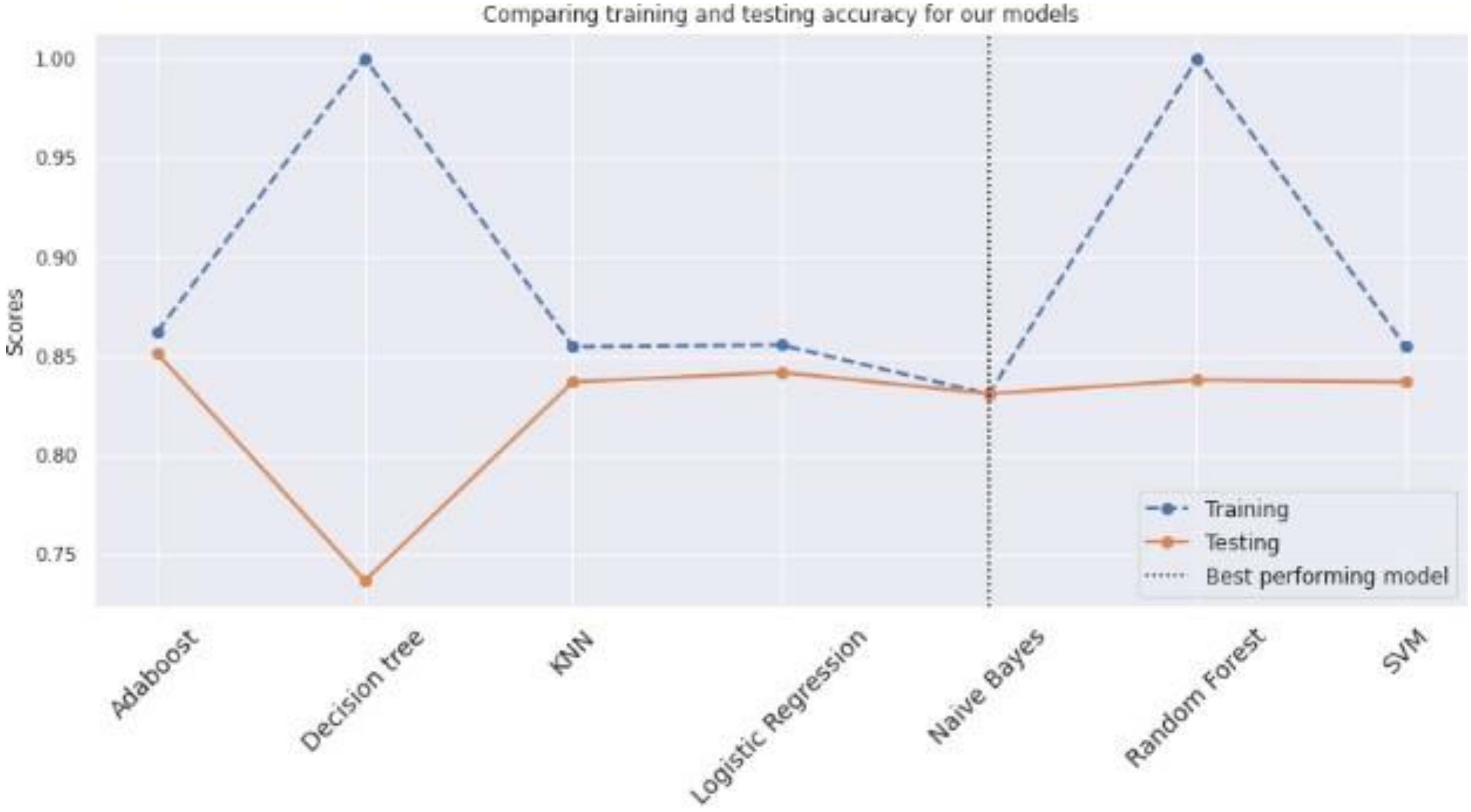
	Name	Train_accuracy	Test_accuracy	Precision	Recall	F1_Score
0	Adaboost	0.861778	0.851524	0.991774	0.854251	0.917890
1	Adaboost after Hyperparameter Tuning	0.856721	0.838741	1.000000	0.838424	0.912111
2	Decision Tree	1.000000	0.740413	0.841363	0.847337	0.844340
3	Decision Tree after Hyperparameter Tuning	0.875685	0.824975	0.958872	0.850886	0.901657
4	KNN	0.854614	0.836775	1.000000	0.836775	0.911135
5	KNN after Hyperparameter tuning	0.868099	0.830875	0.974148	0.846782	0.906011
6	Logistic Regression	0.855457	0.841691	0.998825	0.841584	0.913487
7	Logistic Regression after Hyperparameter Tuning	0.857986	0.848574	0.997650	0.848152	0.916847
8	Naive Bayes	0.830594	0.830875	0.952996	0.860021	0.904125
9	Naive Bayes after Hyperparameter tuning	0.854614	0.836775	1.000000	0.836775	0.911135
10	Random Forest	1.000000	0.838741	0.988249	0.845226	0.911159
11	Random Forest after Hyperparameter Tuning	0.872735	0.840708	0.998825	0.840752	0.912997
12	SVM	0.854614	0.836775	1.000000	0.836775	0.911135
13	SVM after Hyperparameter Tuning	0.860514	0.837758	0.992949	0.841633	0.911051

This is the evaluation metric of all models before and after hyper parameter tuning.

ROC of all the models after Hyperparameter Tuning



Model performance of all models after Hyper parameter tuning



Conclusion

Here after going through the evaluation metric, we can observe that,

(1) To evaluate the model performance, we consider accuracy and recall.

(2) Random Forest, Adaboost and Naïve Bayes are the models which are having high values of recall.

(3) But if we consider accuracy, Random Forest is overfitting. So Adaboost and Naïve Bayes are the models which are performing good.

(4) Among Adaboost and Naïve Bayes, Naïve Bayes is the best performer.

Challenges faced

- The dependent variable has very few data labeled as ‘1’. So we can observe in all models that precision is more. As the models did not had much of two classes ‘0’ and ‘1’ to learn from the data, the models are failing to predict the data as ‘1’.
- Due to this class imbalance, several models are overfitting. Even though we have treated the class imbalance, we could not come over it. We were only able to reduce it.

References

- Analytics Vidhya
- GeeksforGeeks
- Medium

Thank You