Capstone Project

CARDIOVASCULAR RISK PREDICTION

Veeraj

Introduction

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

The classification goal is to predict whether the patient has a 10-year risk of future coronaryheart 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.

Points for discussion

- Data Summary.
- Feature Summary.
- Visualizing Distribution.
- Checking outliers.
- Handling Outliers.
- Cleaning and manipulating the dataset.
- Univariate Analysis and Bivariate analysis.
- Correlation matrix and Handling multicollinearity.
- Model building
  - Logistic Regression
  - Naive Bayes Classifier
  - Support Vector Classifier
  - Random Forest Classifier
  - XGBoost Classifier
  - KNN Classifier
- Conclusion.

Data Summary

```
[ ]  # Check first 5 rows of dataset
     df.head()
```

	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

- This dataset contains 3390 rows and 16 columns.
- Dropping the id column because it just contains unique id number for each patient and will not be used for prediction.
- TenYearCHD our target variable and it consist categorical features and it consist 0 and 1 .

Data Summary

- Missing value count and percentage of each column :
  - glucose (8.97%)
  - education (2.57%)
  - BPMeds (1.30%)
  - totChol (1.12%)
  - cigsPerDay (0.65%)
  - BMI (0.41%)
  - heartRate (0.03%)
- Replacing the NaN values with median, in all the columns.
- Our target variable consist 2879 - 0 and 511 - 1.

Feature Summary

Demographic:

- Sex: male or female("M" or "F"),
- Age: Age of the patient;(Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous).

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).

Feature Summary

- Prevalent Hyp: whether or not the patient was hypertensive (Nominal),
- Diabetes: whether or not the patient had diabetes (Nominal).

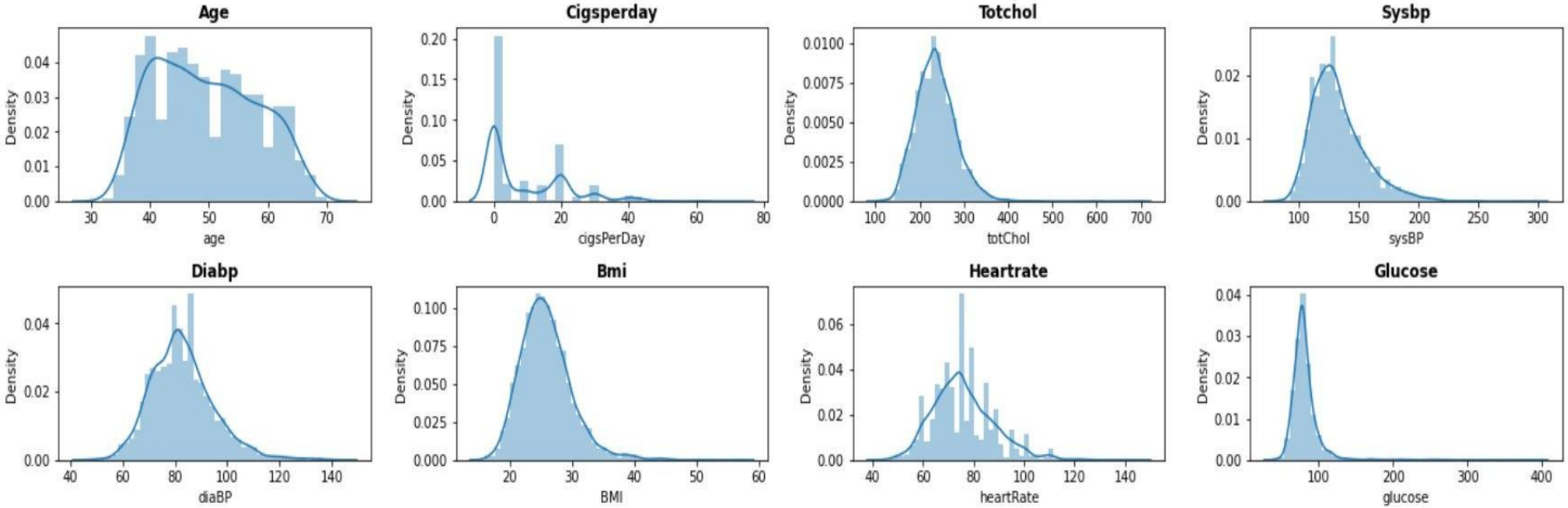
Medical(current)

- 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: heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values),
- Glucose: glucose level (Continuous).

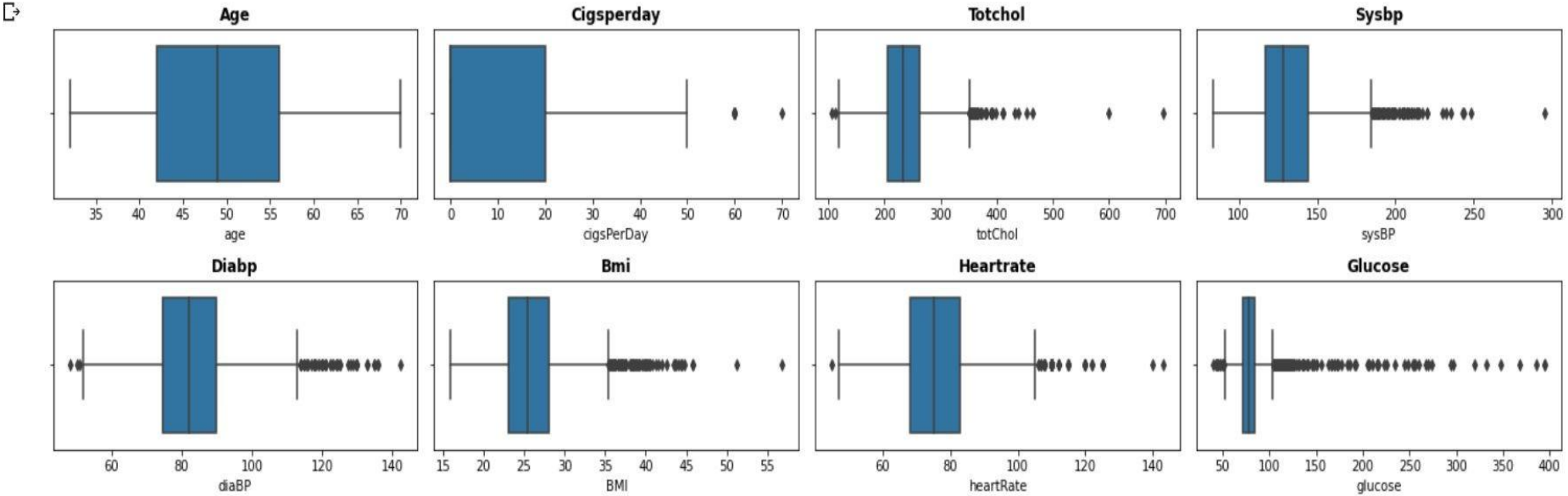
Predict variable (desired target)

- 10-year risk of coronary heart disease CHD(binary: "1", means "Yes", "0" means "No") - DV.

Visualizing Distribution



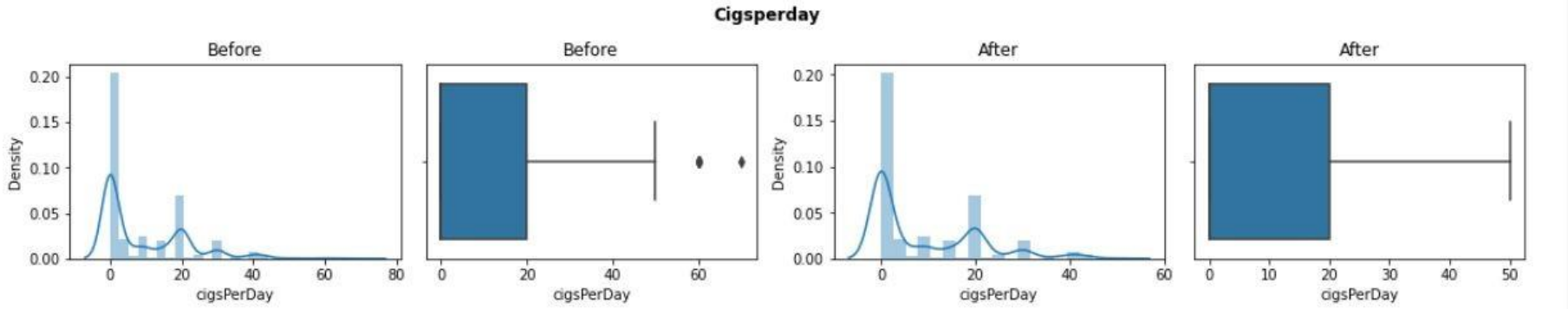
Checking outliers



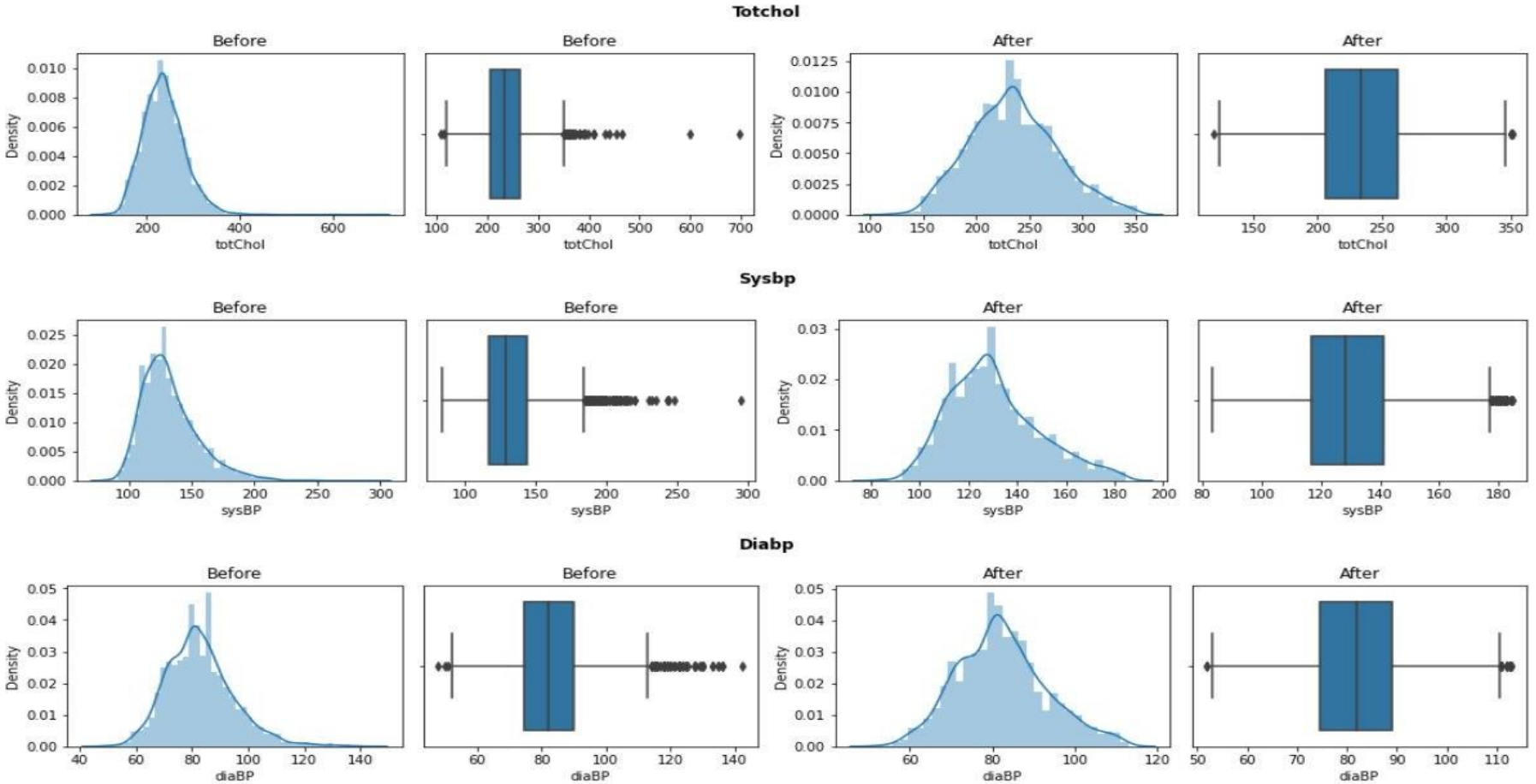
Consider age column remaining all columns are contains outliers so we want treat it with median using interquartile range method(IQR).

Handling Outliers

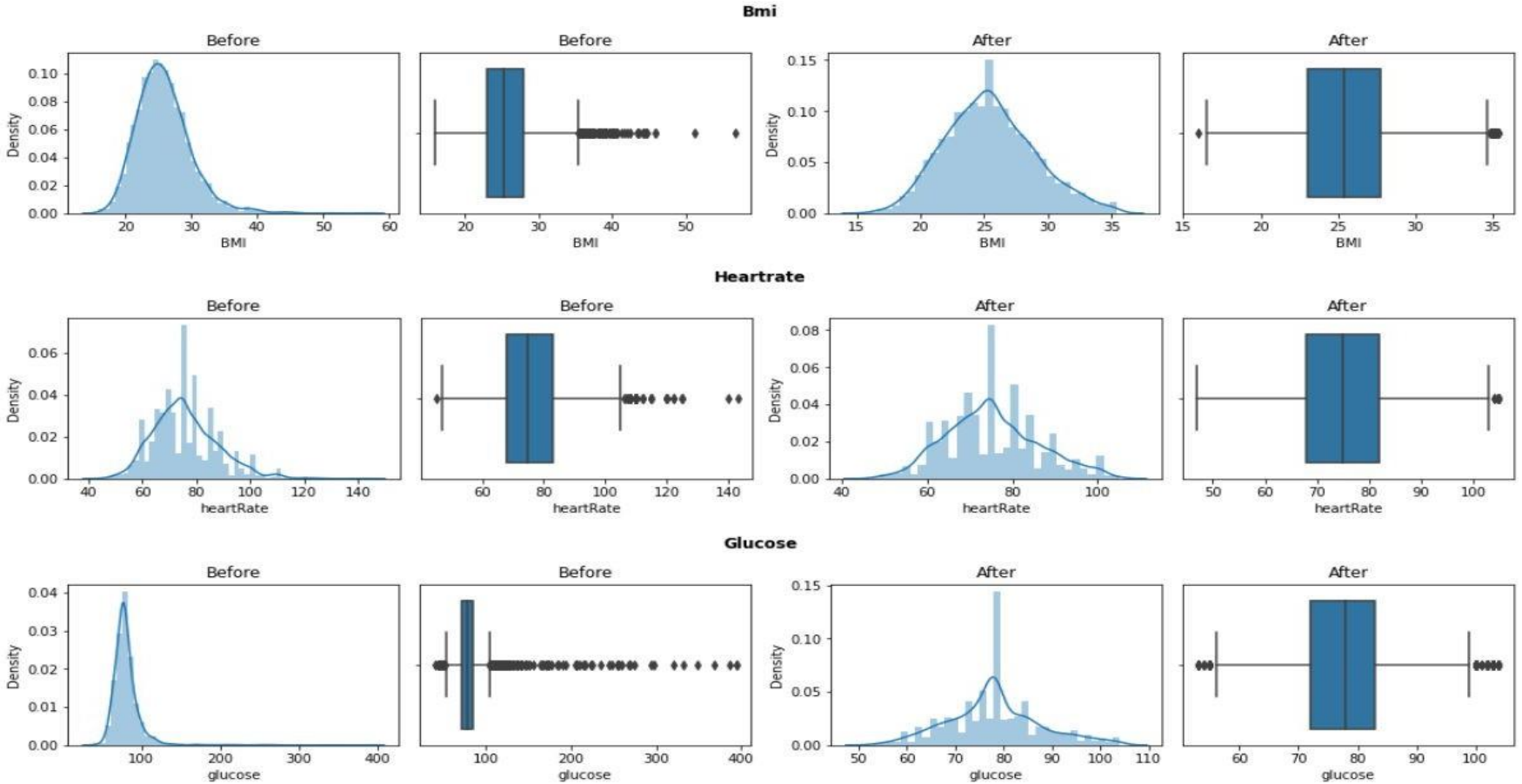
- IQR method of identifying outliers is to set up a “fence” outside of Q1 and Q3, Any values that fall outside of this fence are considered outliers.
- The IQR is the difference between Third quartile and First quartile, To build this fence we take 1.5 times the IQR and then subtract this value from Q1 and add this value to Q3, This gives us the minimum and maximum fence posts that we compare each observation to, Any observations that are more than 1.5 IQR below Q1 or more than 1.5 IQR above Q3 are considered outliers.
- we replaced the outliers with median values i.e. 50th percentile of that column.
- Lets visualize the plots of each feature before and after the outlier treatment.



Handling Outliers



Handling Outliers



Cleaning and Manipulating The Dataset

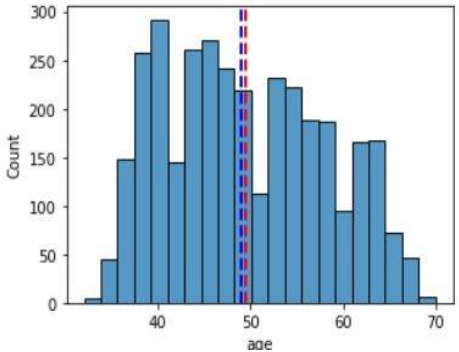
```
for col in ['sex', 'is_smoking']:
  print(data[col].value_counts(),'\n')
```

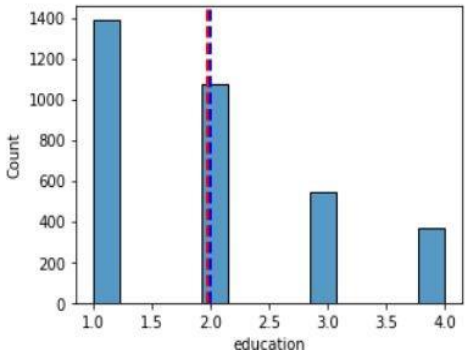
```
F    1923
M    1467
Name: sex, dtype: int64

NO     1703
YES    1687
Name: is_smoking, dtype: int64
```

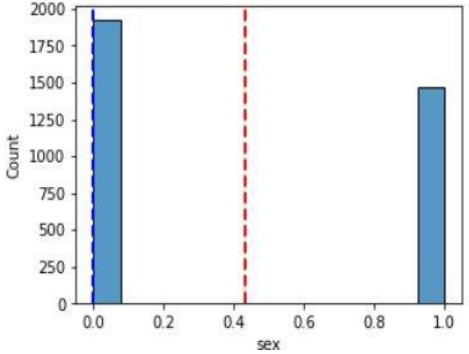
- Checking for the duplicates values in the datasets, showed there are no duplicate records in the dataframe.
- Checking unique value with their counts in categorical features to define an encoder in order to replace those values with numeric values.
- Encoding “M” with 1 and “F” with 0 in the sex column.
- Encoding “YES” with 1 and “NO” with 0 in the is_smoking column.

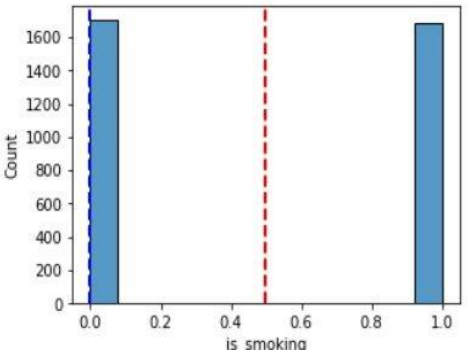
Univariate Analysis

Univariate analysis is to understand the distribution of values for a single variable, It is used to describe the every single feature, Measure the central tendency that means where the mean or median of the dataset is located, and measure the dispersion represent how spread out the values are in the datasets including std deviation and variance.
Red and blue lines in the plot represent the mean and median respectively.





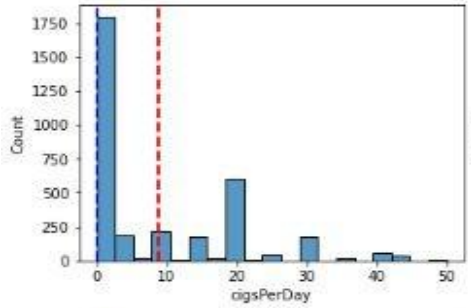


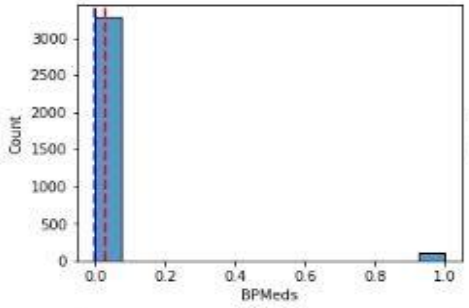


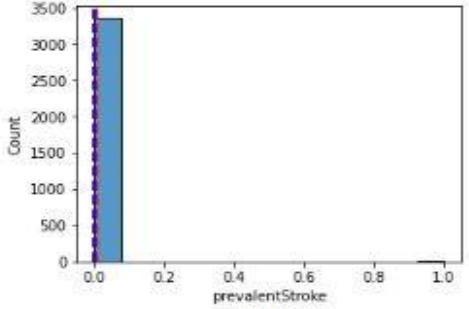
Observation -

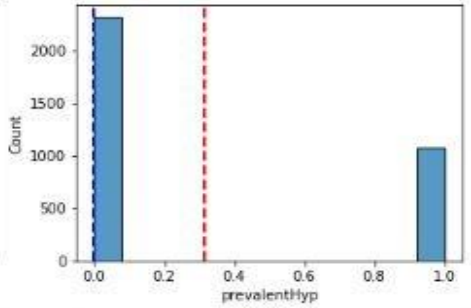
- Most of the people in our dataset are around 40-50 years old,
- Data for Female population is more than the males,
- Most people smoke less than 10 cigarettes a day.

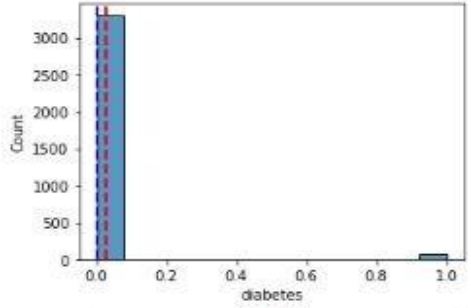
Univariate Analysis

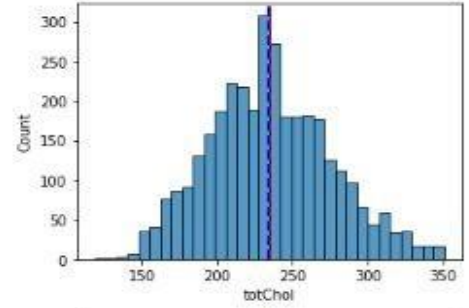


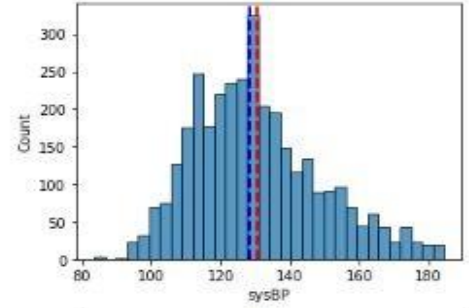


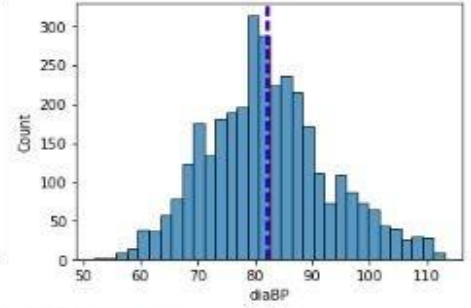


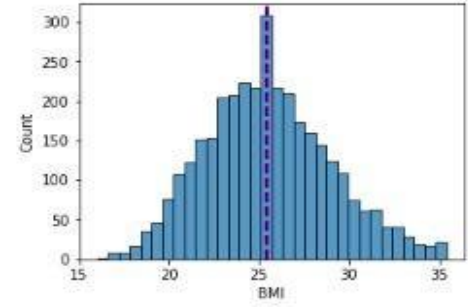


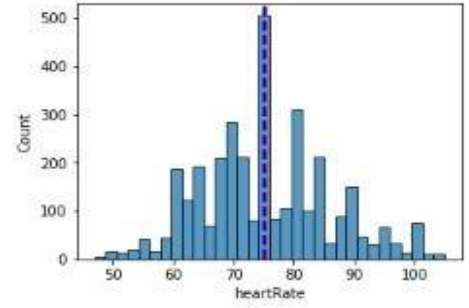


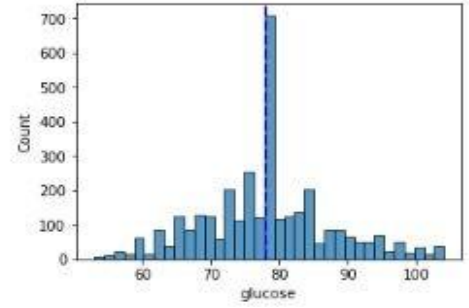


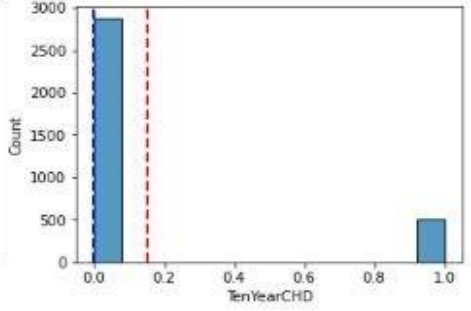






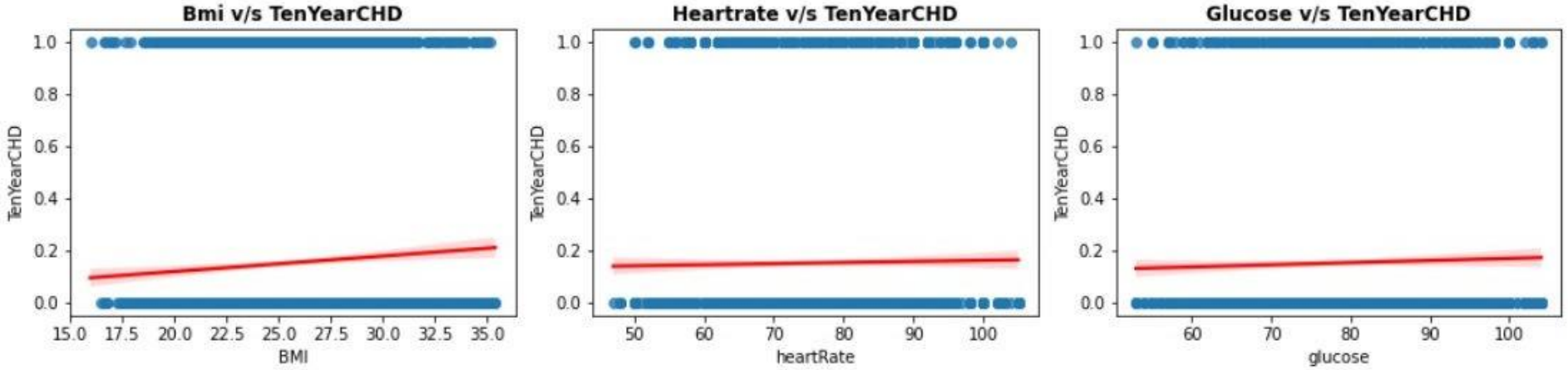






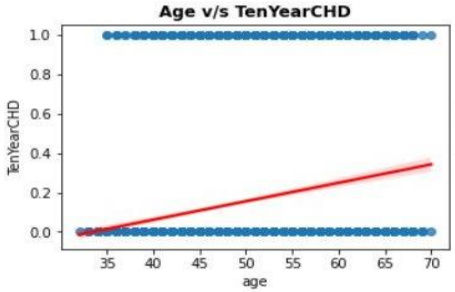
Bivariate Analysis

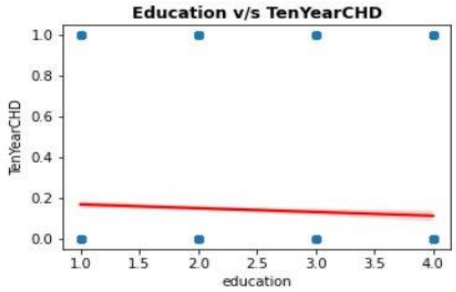
In Bivariate analysis we are visualizing the relation between dependent variable and rest of the independent variable.

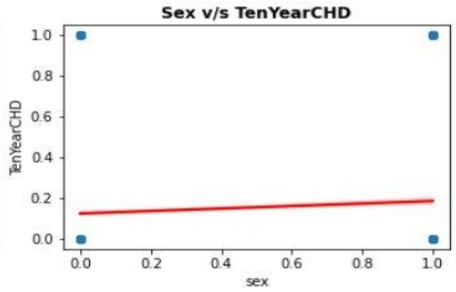


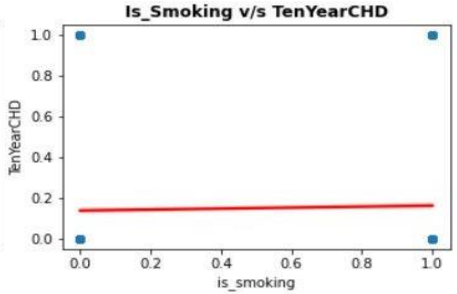
From the plots we can see that age, bmi, totchol, sysbp, diabp etc, are having a clear cut positive relation with the dependent variable, whereas rest of the features have nominal association.

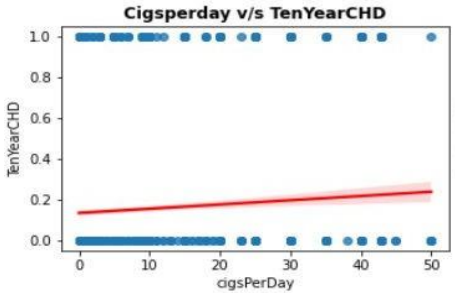
Bivariate Analysis

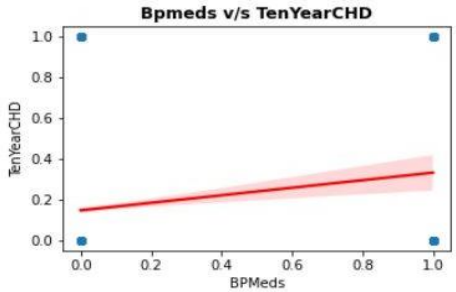


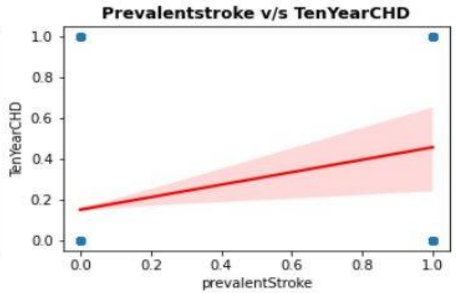


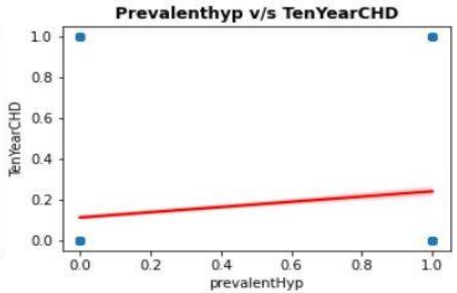


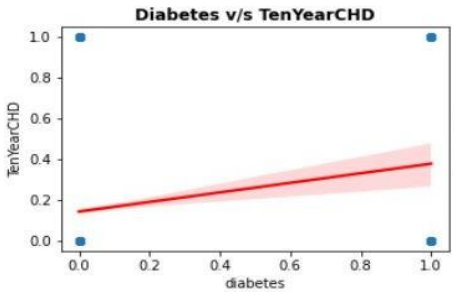


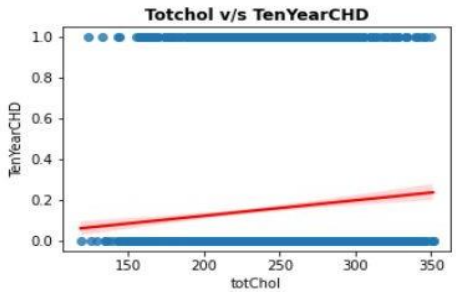


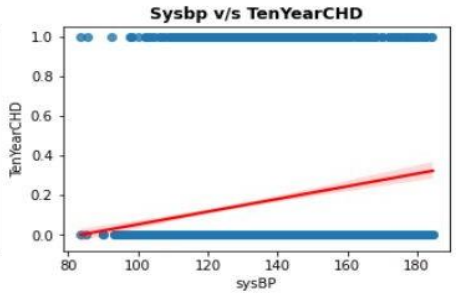


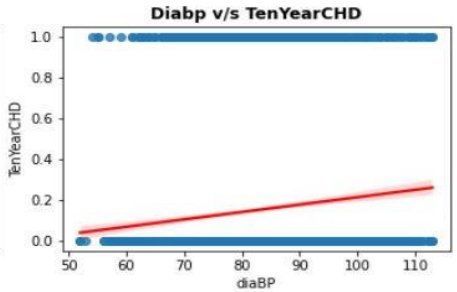




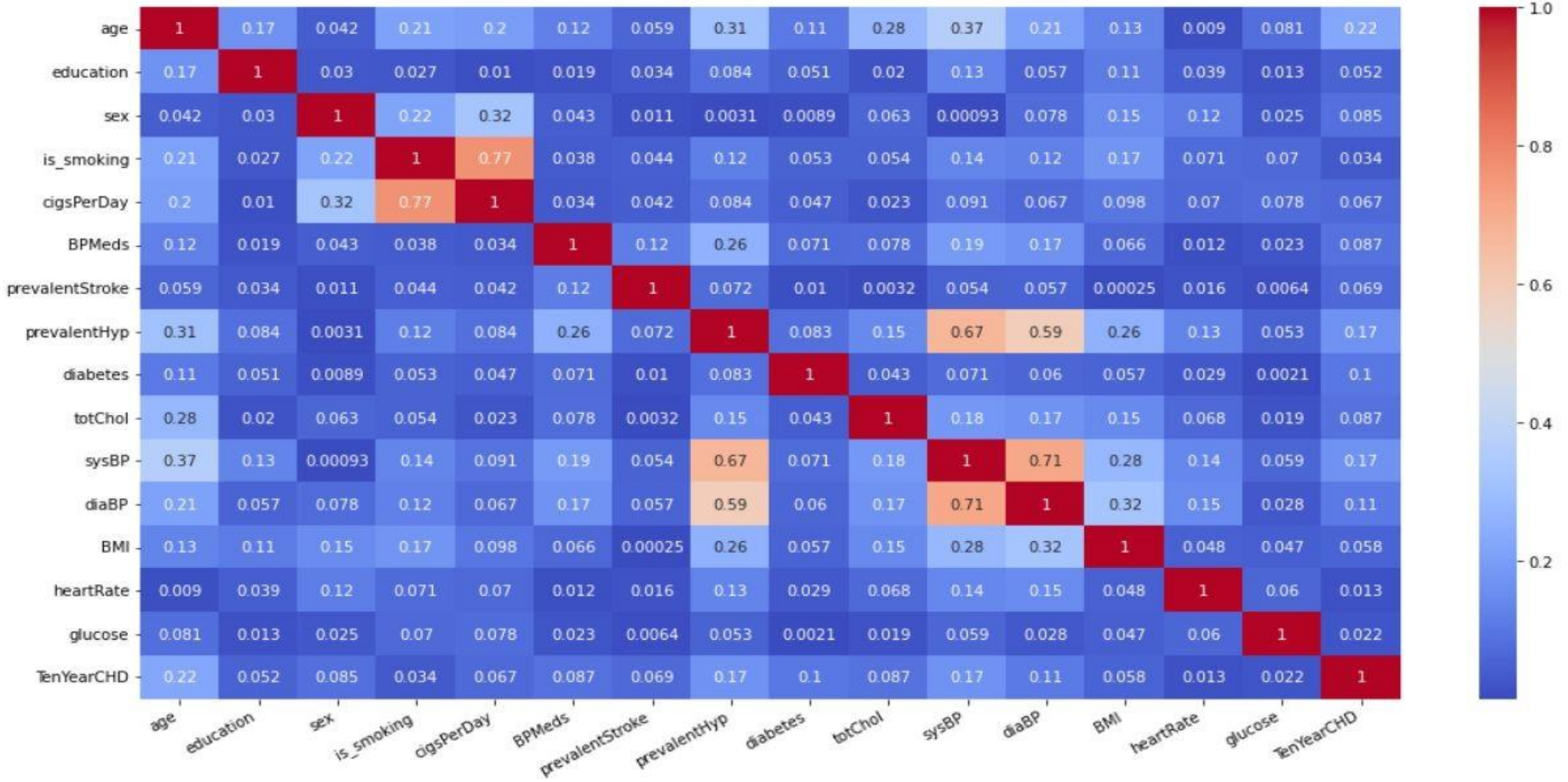








Correlation matrix



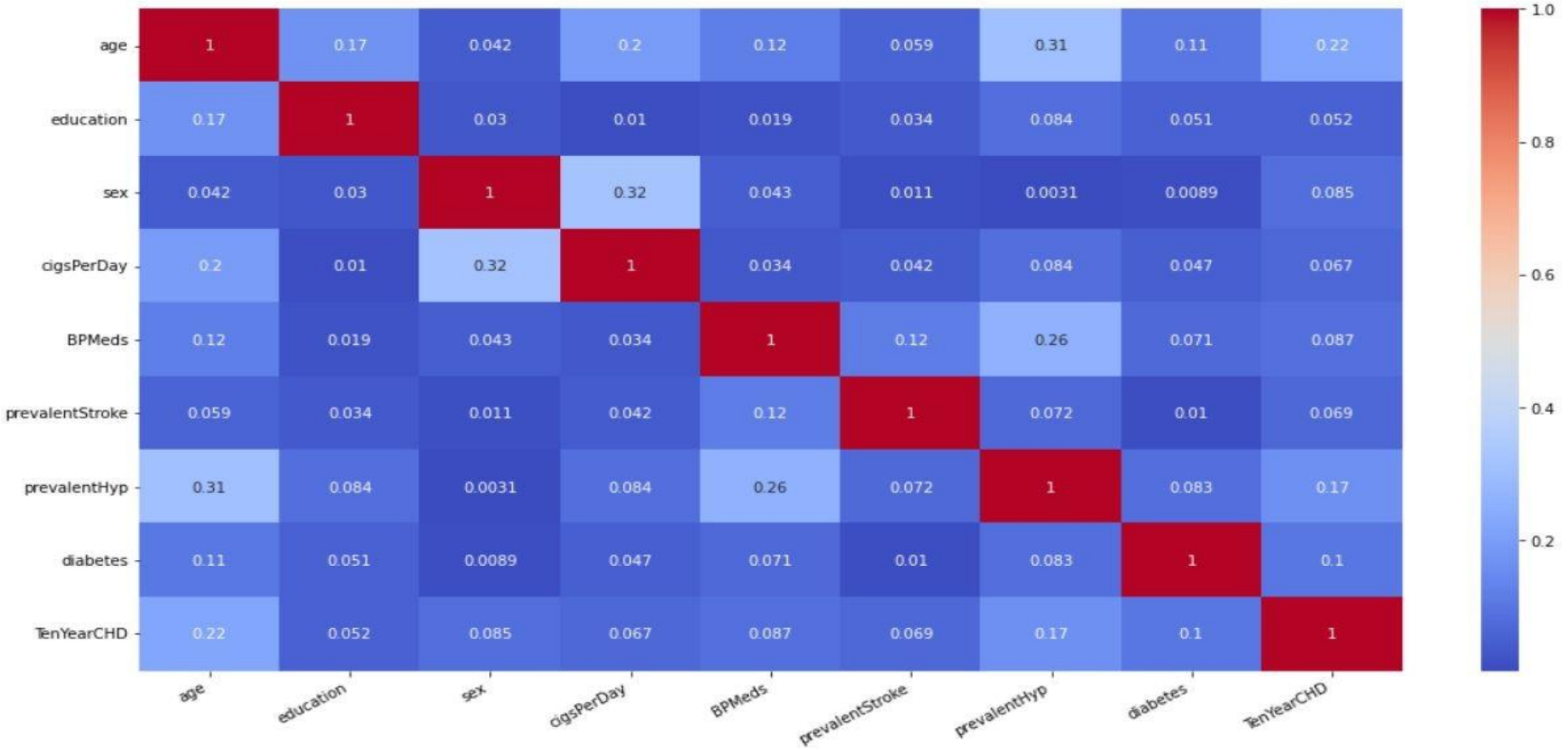
Handling Multicollinearity

	variables	VIF
0	sysBP	132.679399
1	diaBP	127.335444
2	BMI	58.839938
3	glucose	55.695887
4	heartRate	47.760133
5	age	42.764967
6	totChol	37.646845
7	is_smoking	4.955409
8	education	4.831856
9	cigsPerDay	4.195606
10	prevalentHyp	2.359065
11	sex	2.148327
12	BPMeds	1.128283
13	diabetes	1.047201
14	prevalentStroke	1.026839

- **Correlation** tells us how strongly, pairs of variables are related to one another.
- VIF determines the strength of the correlation between independent variables, It is predicted by taking a variable and regressing it against every other variable, VIF score of an independent variable represents how well the variable is explained by other independent variables.
- we have excluded the features whose VIF score is higher than 10. Pictures in the left and right shows the VIF scores of variables before and after multicollinearity treatment.

	variables	VIF
0	age	5.513455
1	education	4.100370
2	sex	1.968156
3	cigsPerDay	1.733136
4	prevalentHyp	1.686226
5	BPMeds	1.120401
6	diabetes	1.044716
7	prevalentStroke	1.024945

Updated Heatmap



Model Building Prerequisites

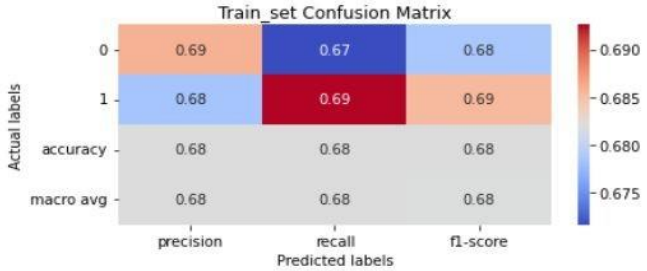
Before fitting the model we has to normalize our dataset.

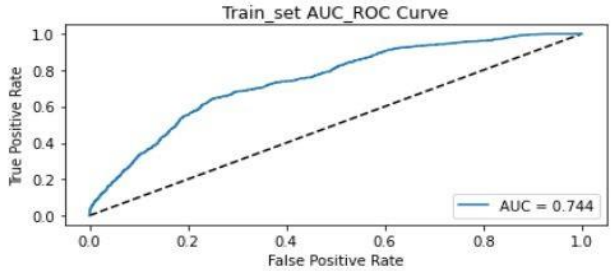
- Here we using MinMaxScaler for scaling the features.
- Making a variable to define F1 score of class 1 of the target variable so as to use it at the time of hyperparameter tuning because by default Gridsearch will maximize the Macro Average of F1 score for all classes. However we want to maximize the F1 score of class 1.
- Defining X and Y variables, and splitting the data in 80 20 ratio as train and test sets.
- Handling class imbalance by oversampling using SMOTE followed by removing the Tomek links. Finally Checking value counts for both classes Before and After handling Class Imbalance.

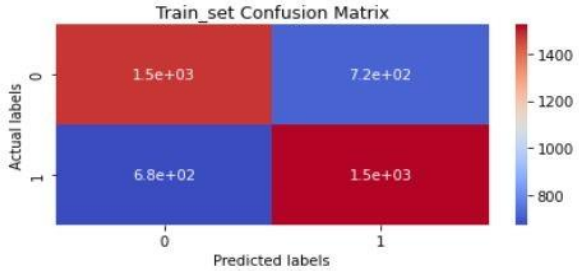
```
BeforeHandling Class Imbalance:
0    2305
1     407
Name: TenYearCHD, dtype: int64

AfterHandling Class Imbalance:
0    2199
1    2199
Name: TenYearCHD, dtype: int64
```

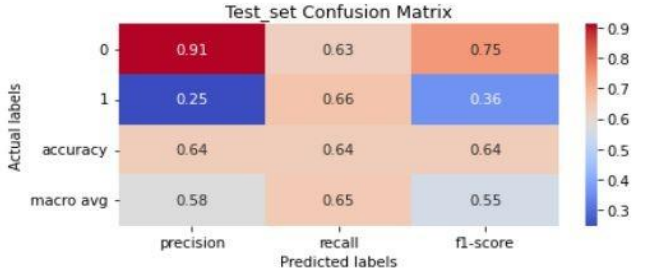
Defining a function which takes classifier model and train test splits as input and outputs the classification report for model performance on train and test data. Also plots the feature importance.

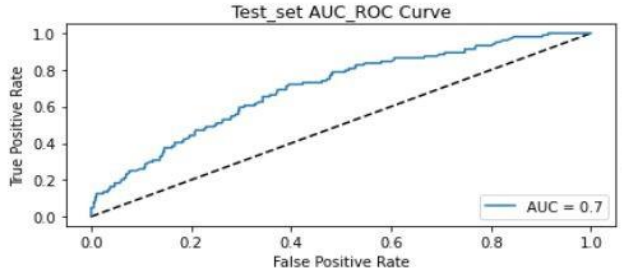
Logistic Regression

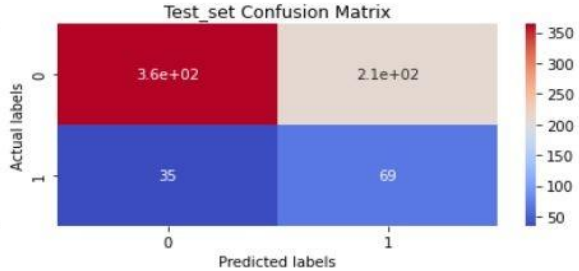


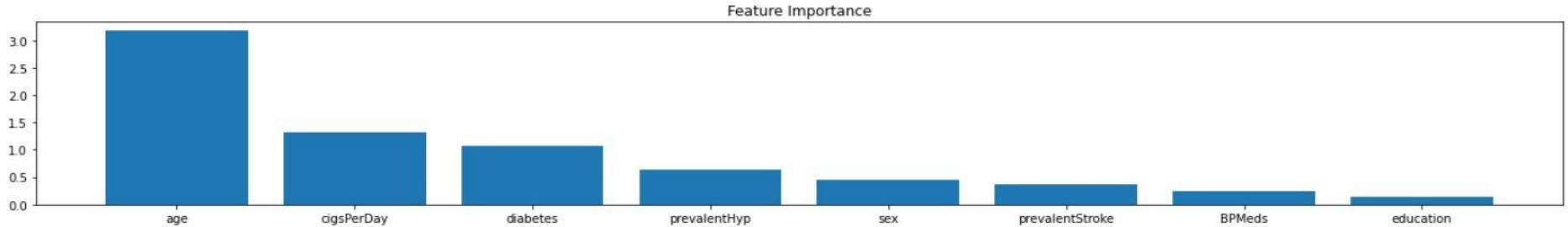










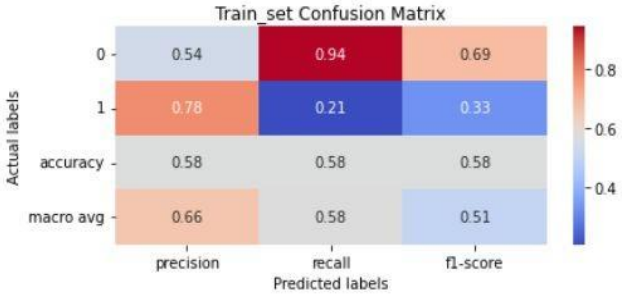


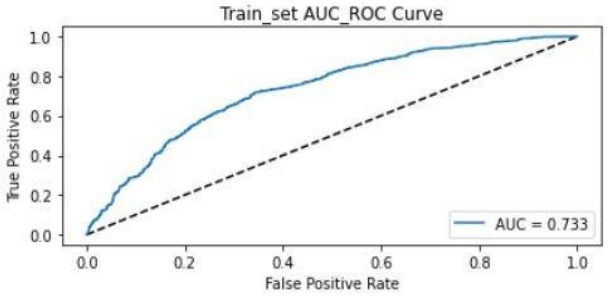

```
LogisticRegression(max_iter=10000)
```

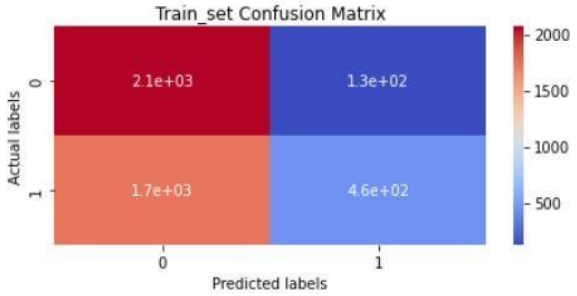
Logistic Regression

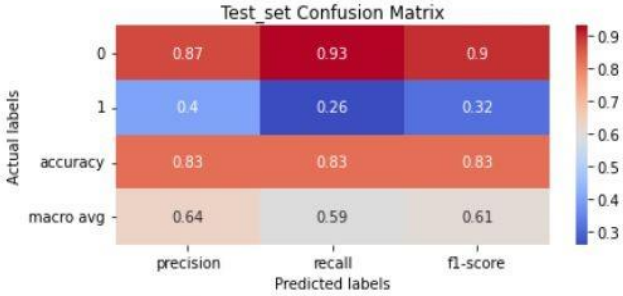
- Starting with the quick and dirty models first, then proceeding towards the complex models. Logistic regression outputs following result for class 1 on test data:
  - Precision 0.25
  - Recall 0.66
  - F1 Score 0.36
- The feature importance plotted is based on the beta coefficients of z (i.e. before applying sigmoid function).
- Age is the most influencing feature, followed by CigsPerDay followed by diabetes.

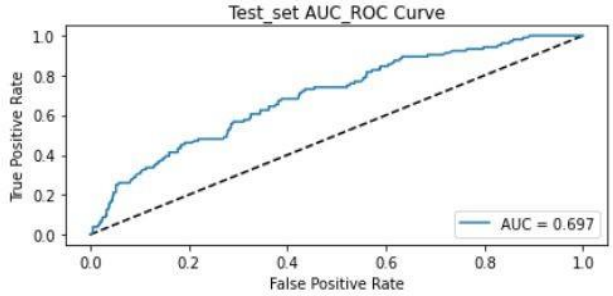
Naive Bayes Classifier

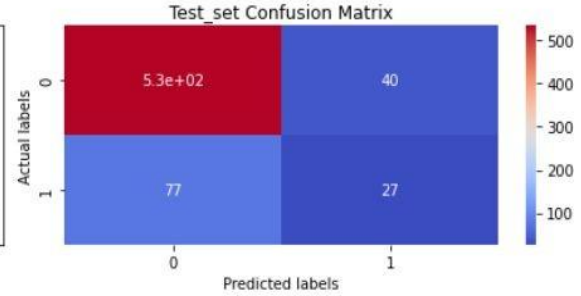










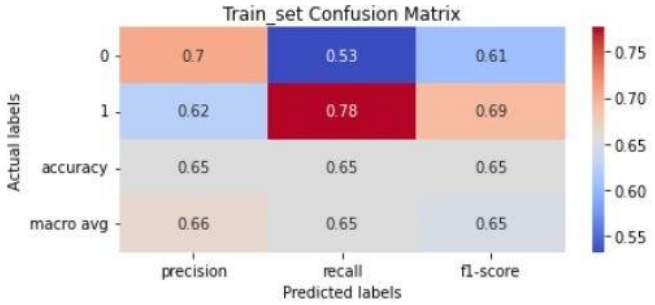


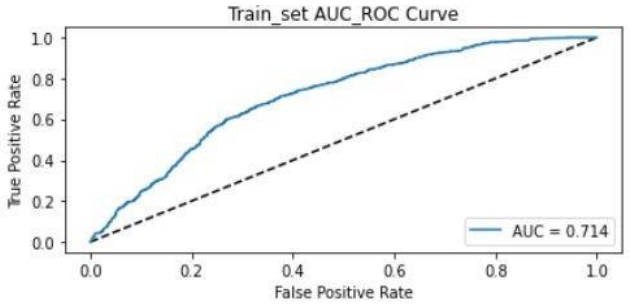

```
<Figure size 1296x216 with 0 Axes>
GaussianNB()
```

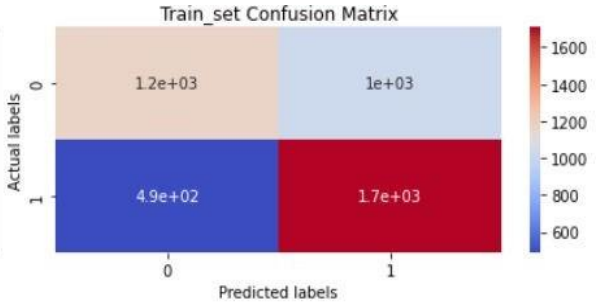
Naïve Bayes Classifier is very fast to implement and may be used as a baseline model to compare with different models. It outputs following result for class 1 on test data,

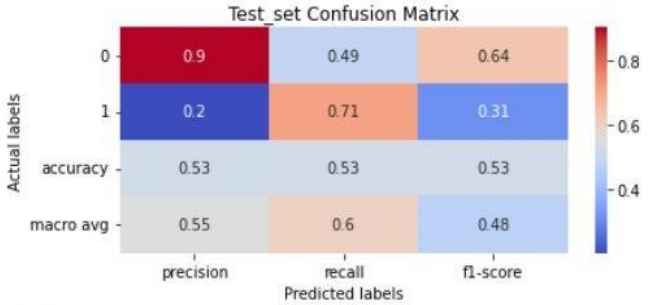
- Precision 0.4
- Recall 0.26
- F1 Score 0.32

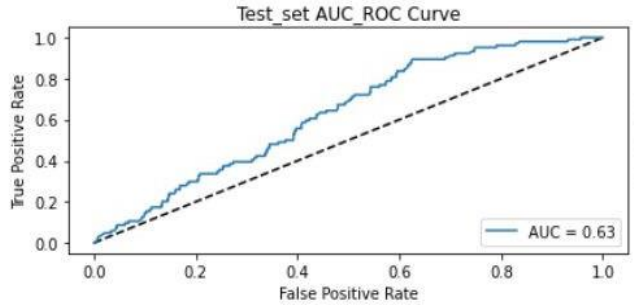
Support Vector Classifier

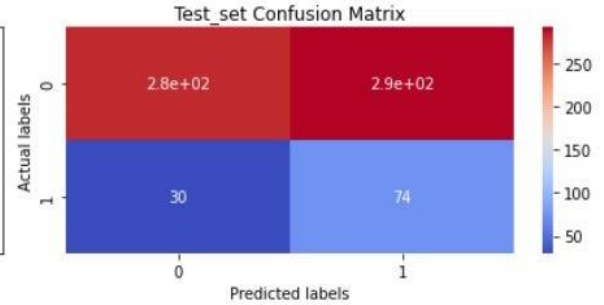










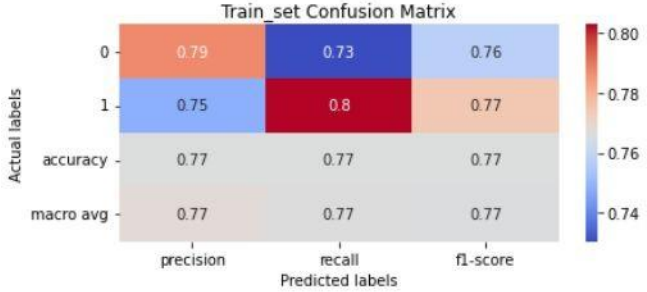


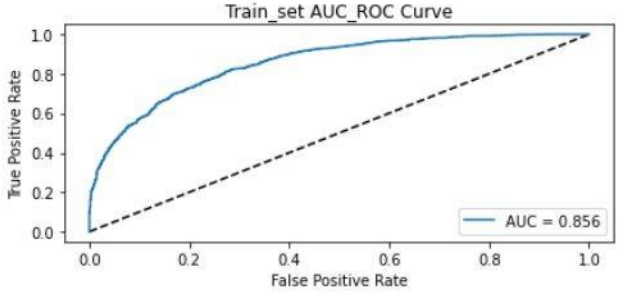

```
<Figure size 1296x216 with 0 Axes>
SVC(C=0.1, max_iter=1000, probability=True, random_state=0)
```

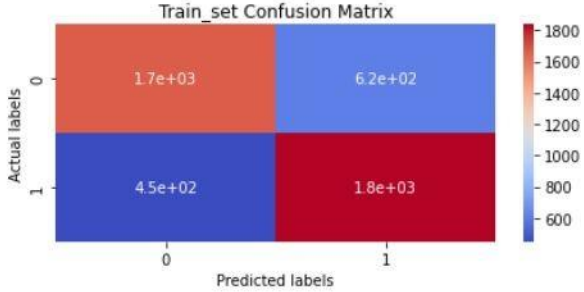
Support Vector Classifier with C= 0.1 outputs following result for class 1 on test data,

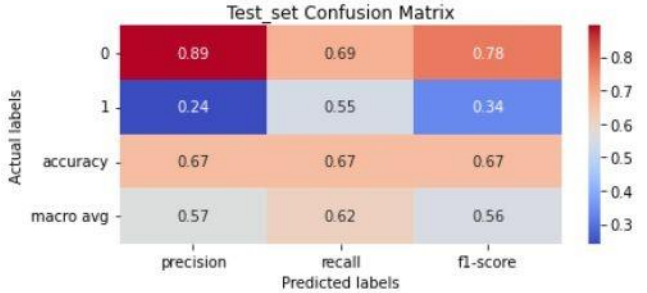
- Precision 0.2
- Recall 0.71
- F1 Score 0.31

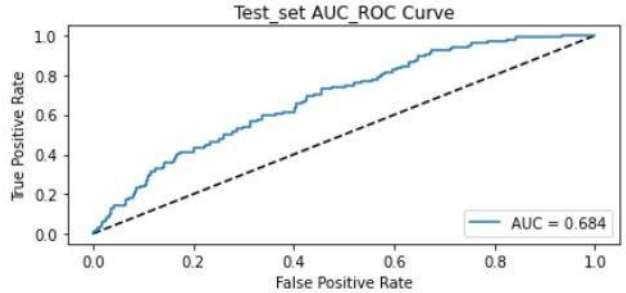
Random Forest Classifier

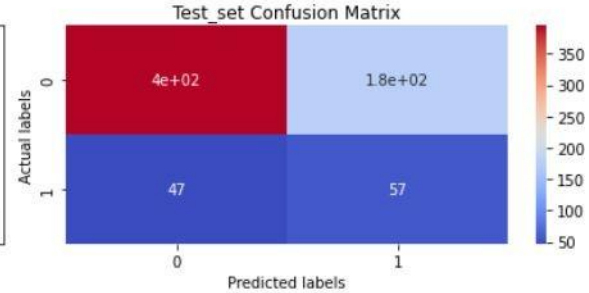










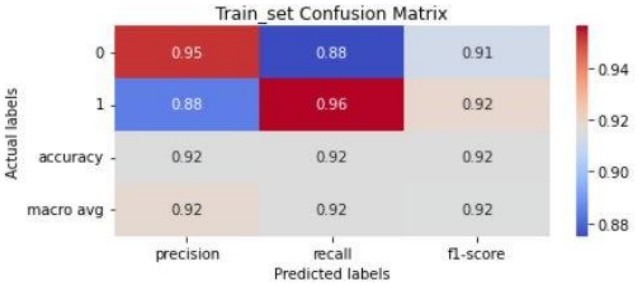


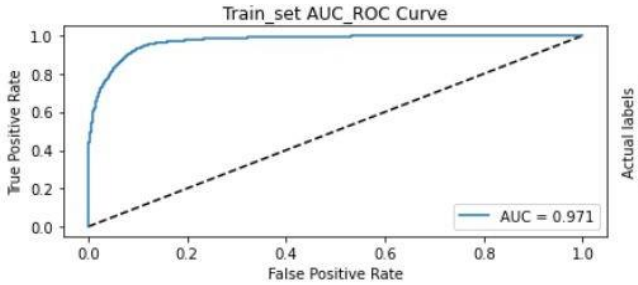

```
<Figure size 1296x216 with 0 Axes>
RandomForestClassifier(max_depth=8, min_samples_leaf=46, min_samples_split=50,
                       random_state=2)
```

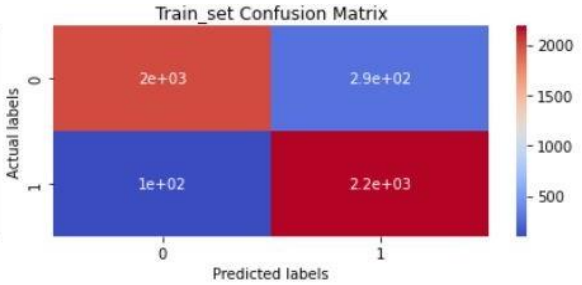
RandomForestClassifier(max_depth=8, min_samples_leaf=46, min_samples_split=50) gives following result for class 1 on test data:

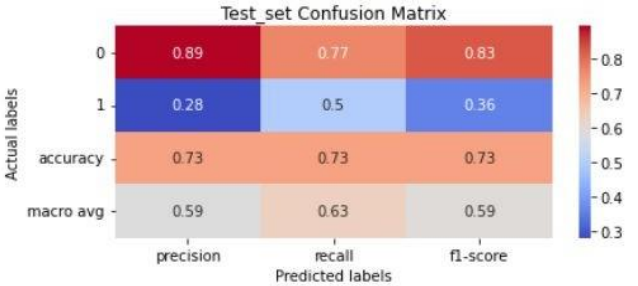
- Precision 0.24
- Recall 0.55
- F1 Score 0.34

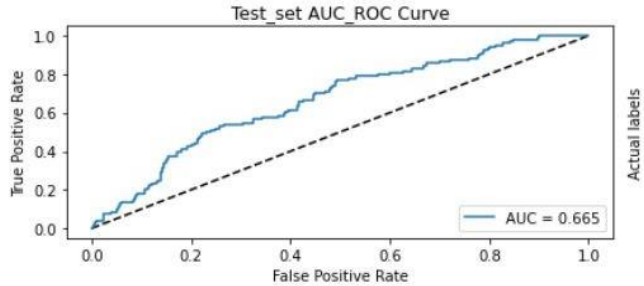
XGBoost Classifier

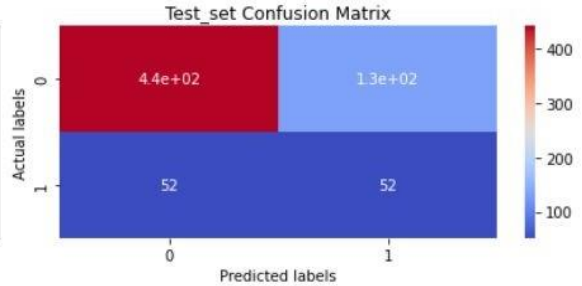










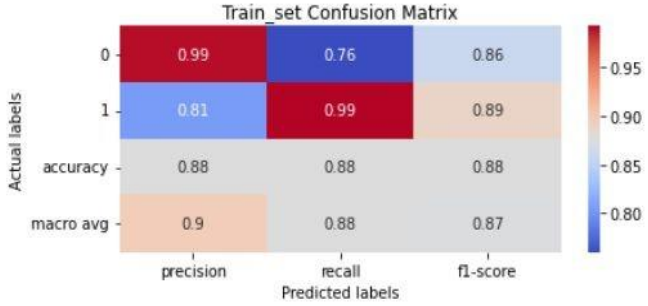


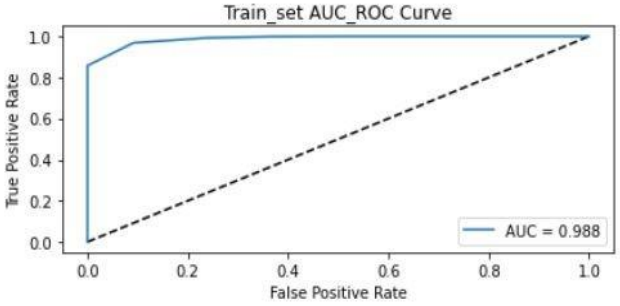

```
<Figure size 1296x216 with 0 Axes>
XGBRFClassifier(eta=0.05, max_depth=10, min_samples_lead=30,
                min_samples_split=50, n_estimators=150, random_state=3,
                silent=True)
```

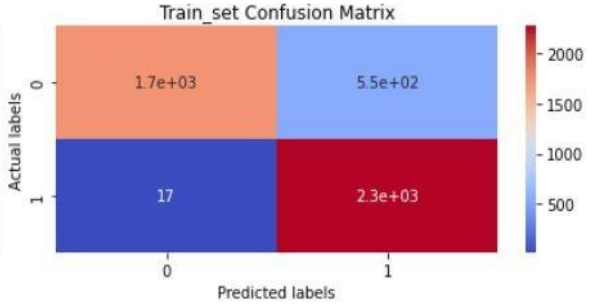
XGBClassifier(eta=0.05, max_depth=10, min_samples_leaf=30, min_samples_split= 50,n_estimators=150) gives following result for class 1 on test data:

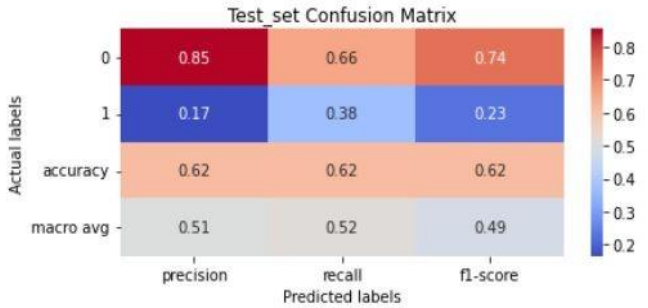
- Precision 0.28 and Recall 0.5,
- F1 Score 0.36

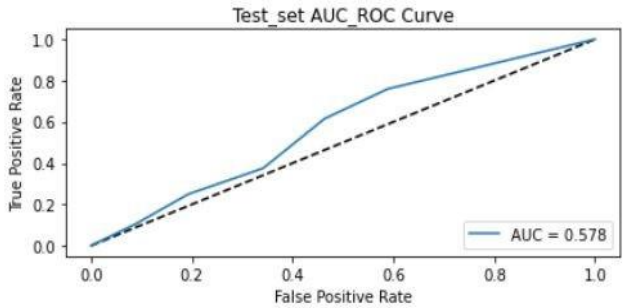
KNN Classifier

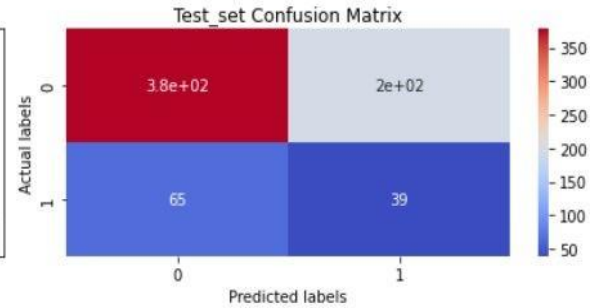













```
<Figure size 1296x216 with 0 Axes>
KNeighborsClassifier(metric='manhattan')
```

KNeighborsClassifier(metric='manhattan', 'n_neighbors=5) gives following result for class 1 on test data:

- Precision 0.17
- Recall 0.38
- F1 Score 0. 23

Conclusion

- If we want to completely avoid any situations where the patient has heart disease, a high recall is desired. Whereas if we want to avoid treating a patient with no heart diseases a high precision is desired.
- Assuming that in our case the patients who were incorrectly classified as suffering from heart disease are equally important since they could be indicative of some other ailment, so we want a balance between precision and recall and a high f1 score is desired.
- Since we have added synthetic data points to handle the huge class imbalance in training set, the data distribution in train and test are different so the high performance of models in the train set is due to the train test data distribution mismatch and not due to overfitting.
- Best performance of Models on test data based on evaluation metrics for class 1,
  - Recall - SVC2.
  - Precision - Naive Bayes Classifier
  - F1 Score - Logistic Regression, XGBoost
  - Accuracy - Naive Bayes Classifier

Thank You

Veeraj