Capstone Project-III

Cardiovascular Risk prediction

Team Members

Arun Teja Lonka
Zunaid
Upasana Kumari
Sukesh Shetty

Game Plan

AI

- Introduction
- Exploratory Data Analysis
- Feature Processing
- ML Model development
- Results & Conclusions
- Challenges Faced

Introduction

- Cardiovascular Heart diseases (CHDs) are the leading cause of death globally.

- An estimated 17.9 million people died from CHDs in 2019, representing 32% of all global deaths. Of these deaths, 85% were due to heart attack and stroke.

- Most cardiovascular diseases can be prevented by addressing behavioural risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol.

Problem Statement

AI

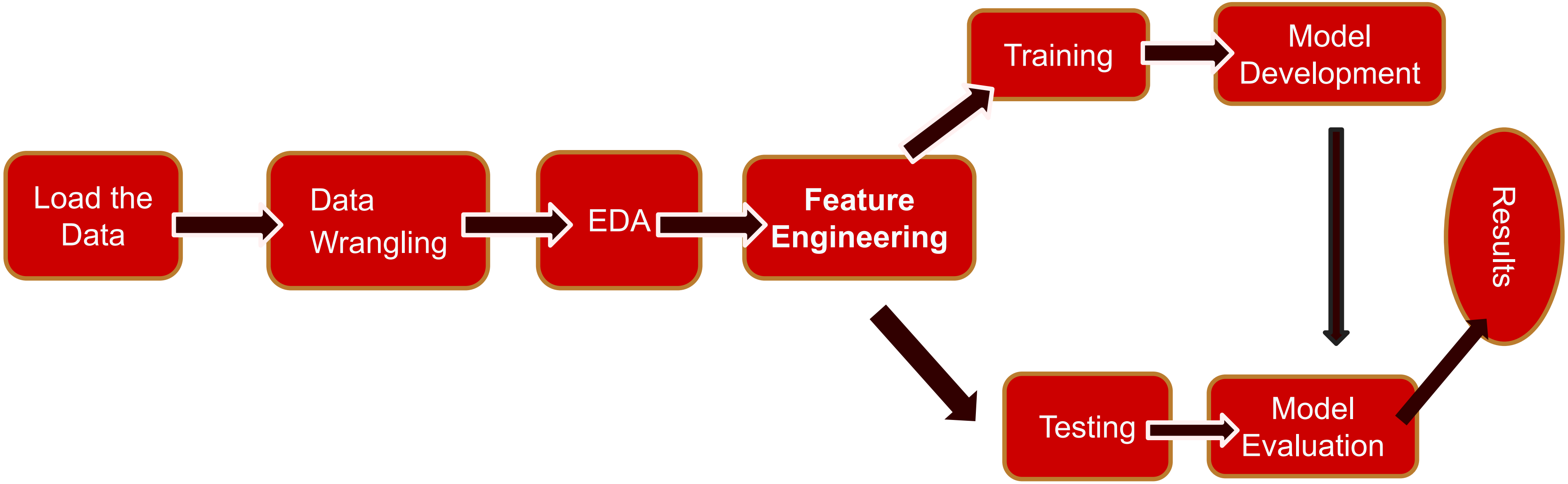
- The dataset is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD). The dataset provides the patients' information. It includes over 4,000 records and 15 attributes.

- Numerical columns: id, age, totChol, sysBP, diaBP, BMI, heart rate, glucose

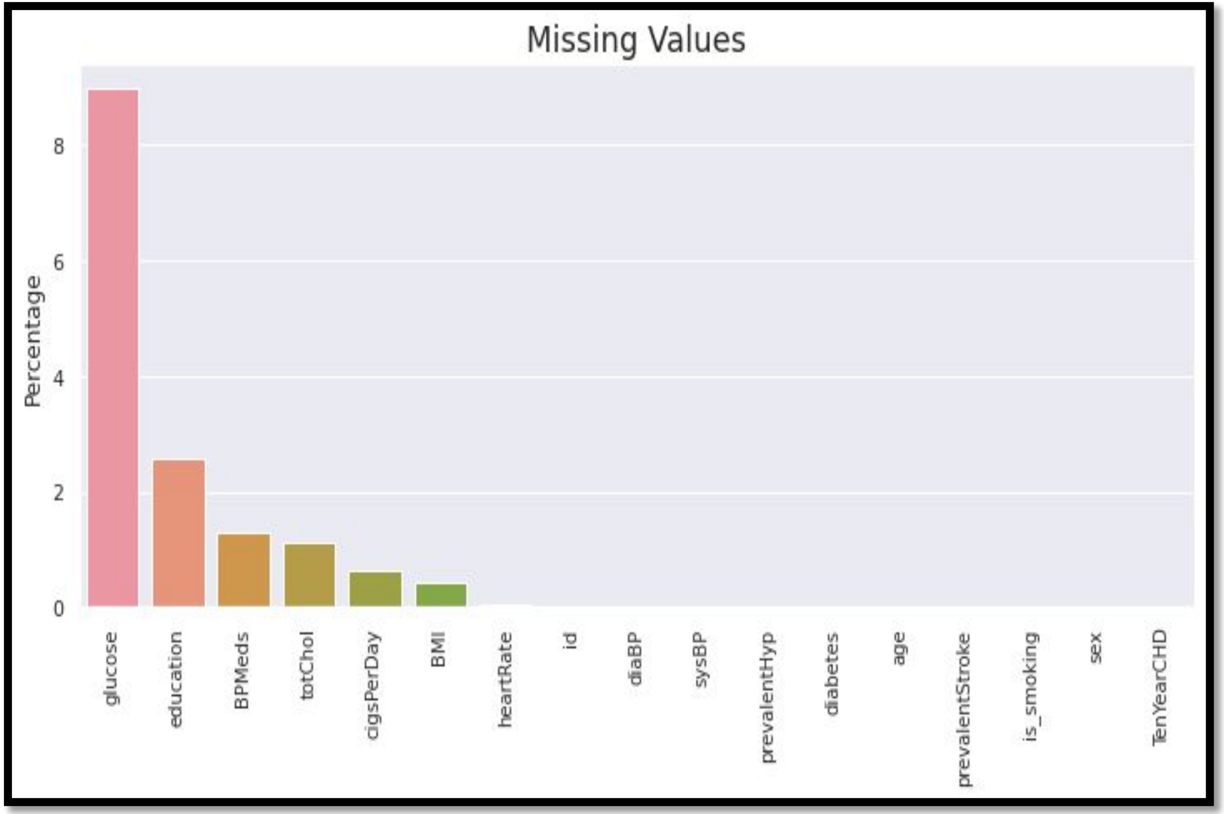
- Categorical columns: education, cigsPerDay, sex, is smoking, BPMed, prevalent Stroke, prevalentHyp, diabetes, TenYearCHD.

- Dataset shape: 3390 Rows and,17 Columns/Features

Flow of the Project

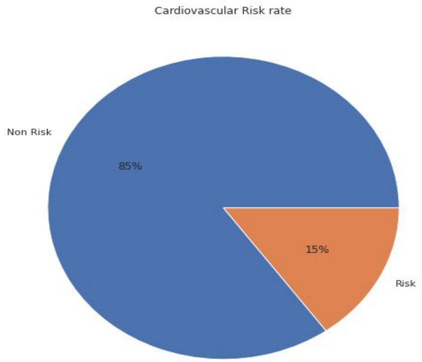


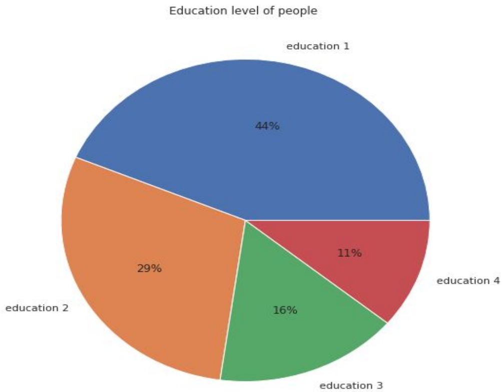
Missing values

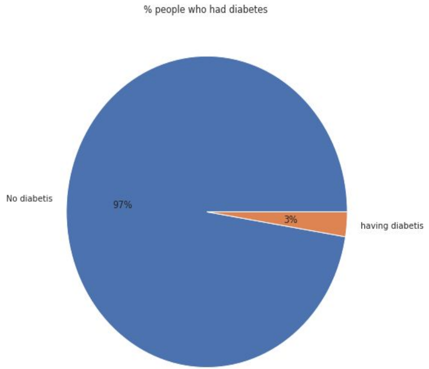


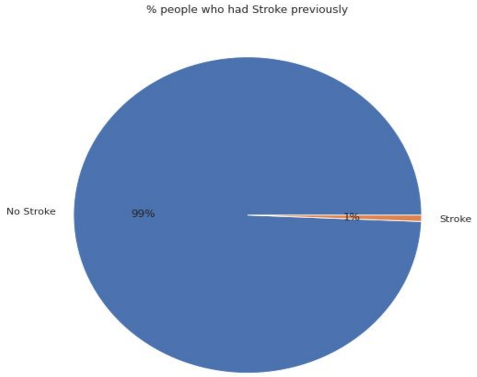
Column	Total	Percentage
glucose	304	8.97
education	87	2.57
BP Meds	44	1.30
totChol	38	1.12
cigsPerDay	22	0.65
BMI 14	0.41	
heart rate	1	0.03

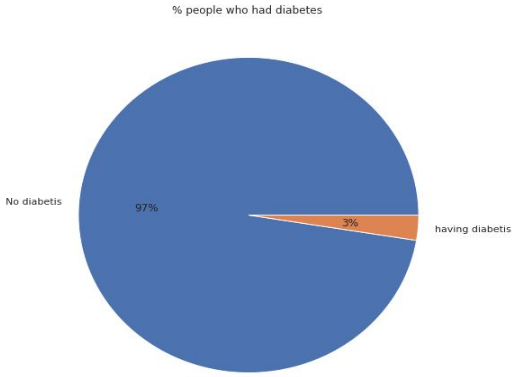
Univariant Analysis

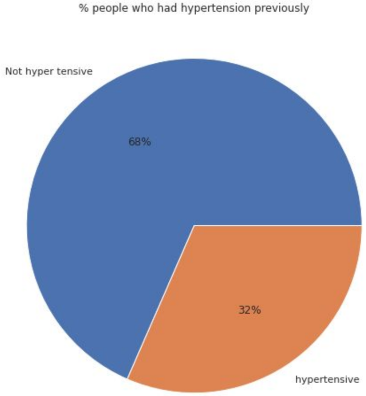




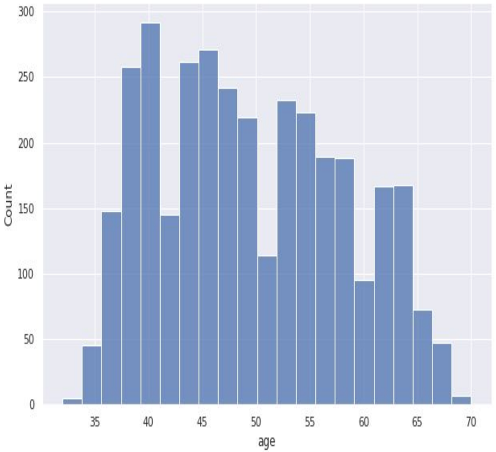


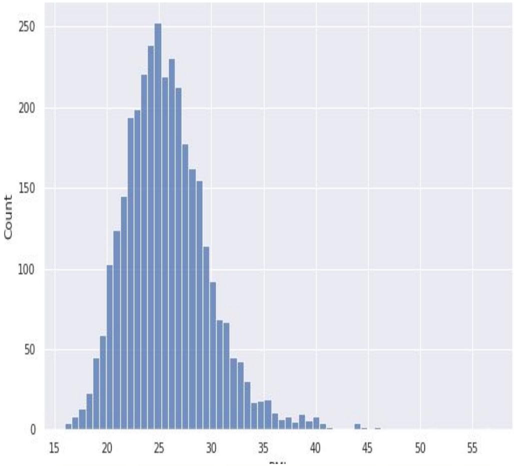


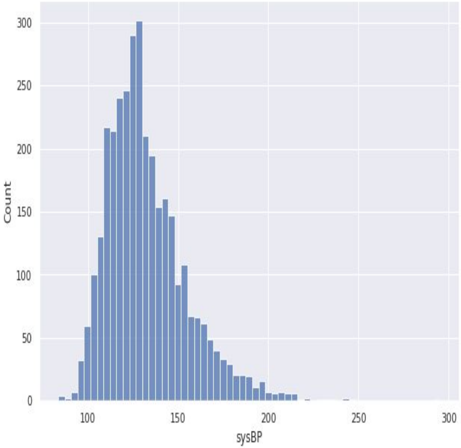


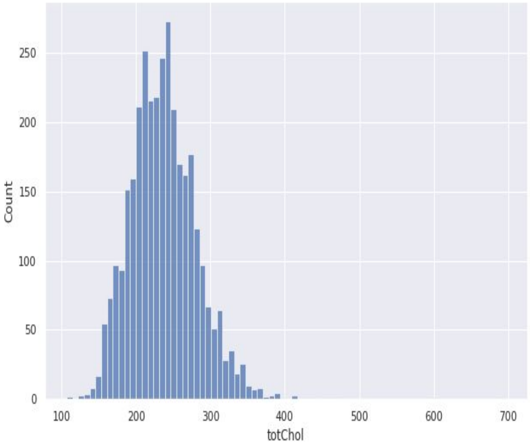


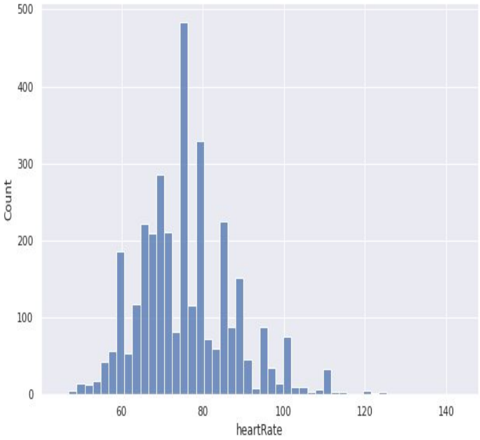
Univariate Analysis

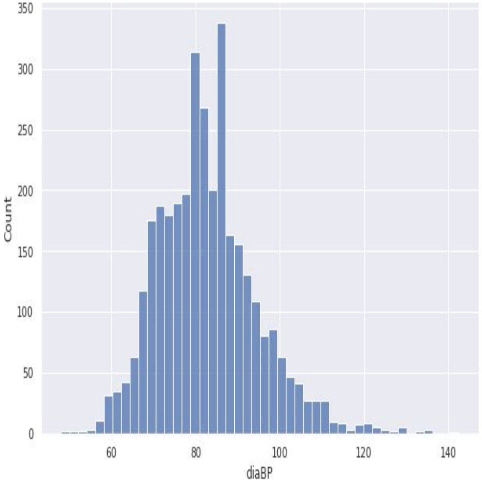








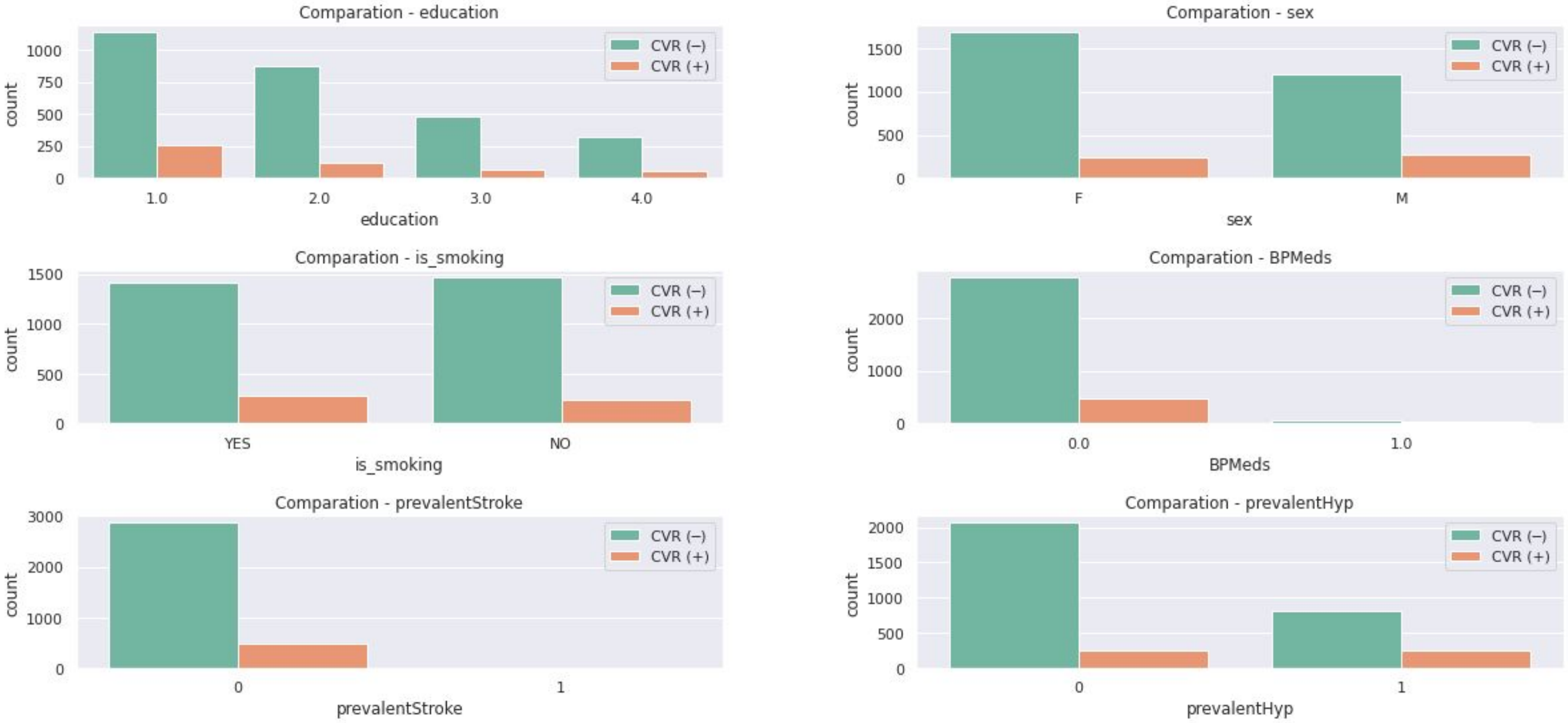




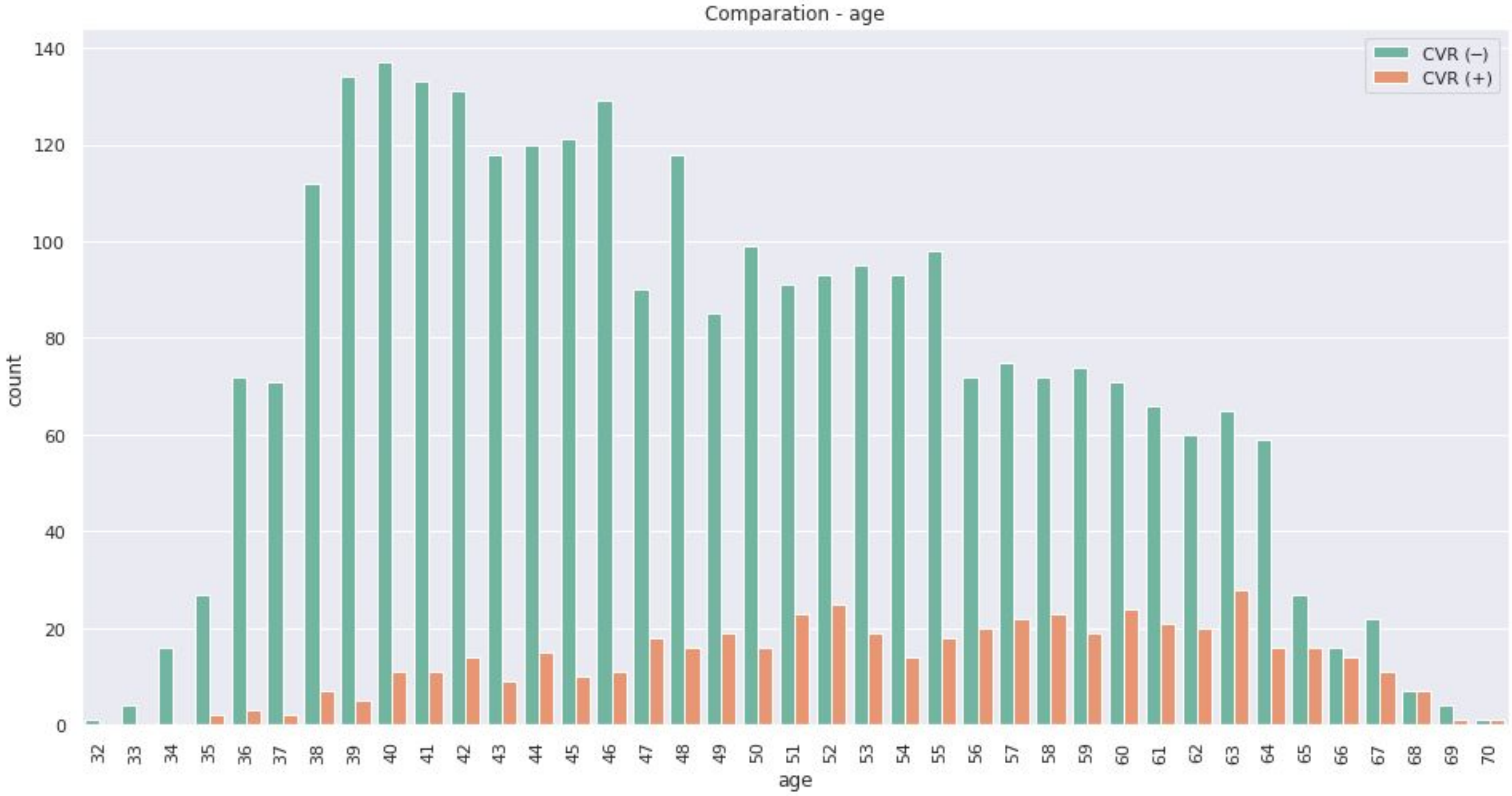
Bi-variant analysis

AI

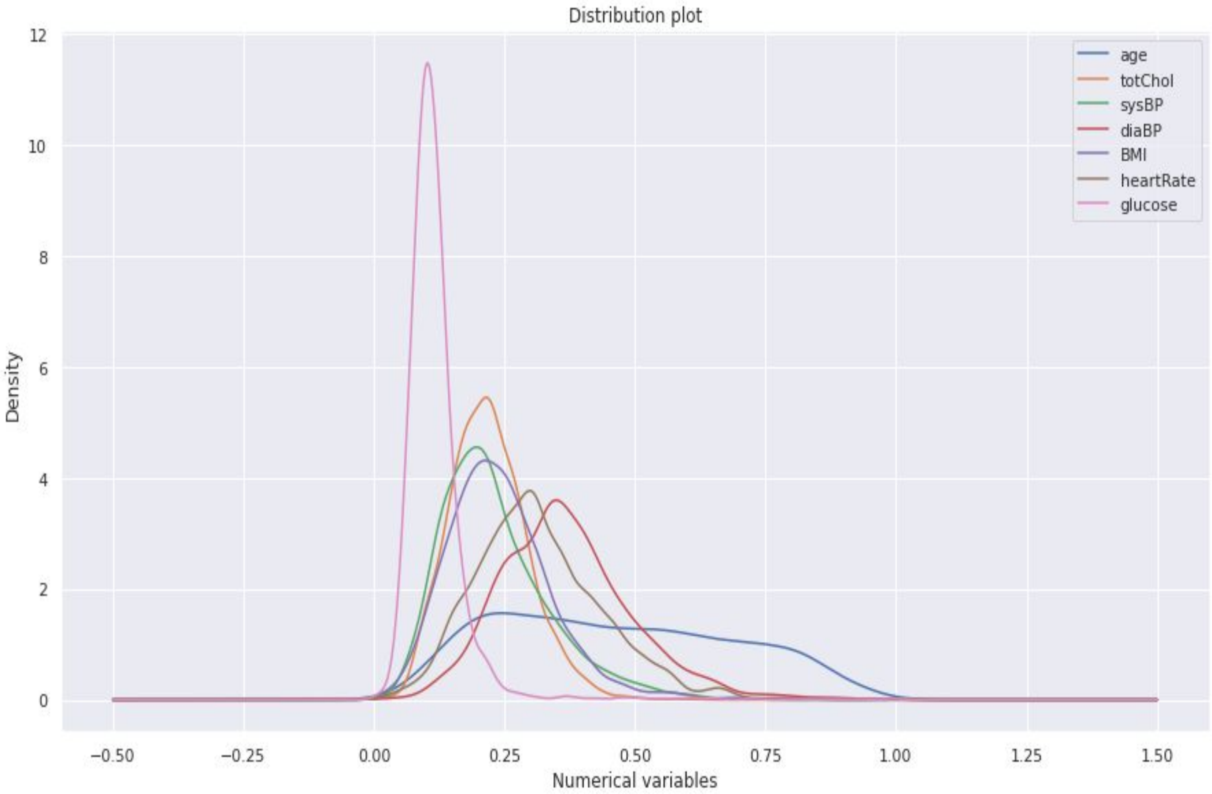
CVR -> Cardio Vascular Risk Disease



Bi-variant analysis

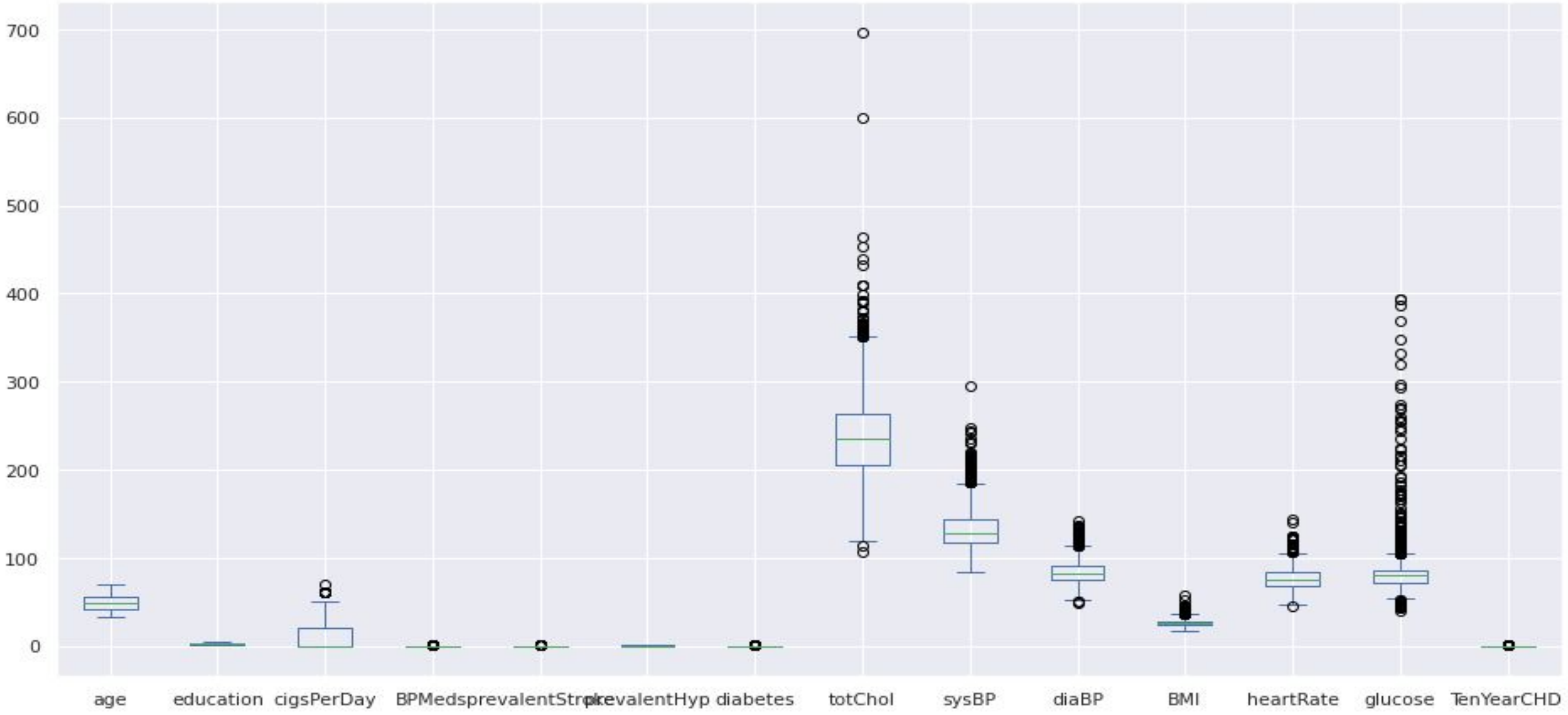


Conclusions of EDA



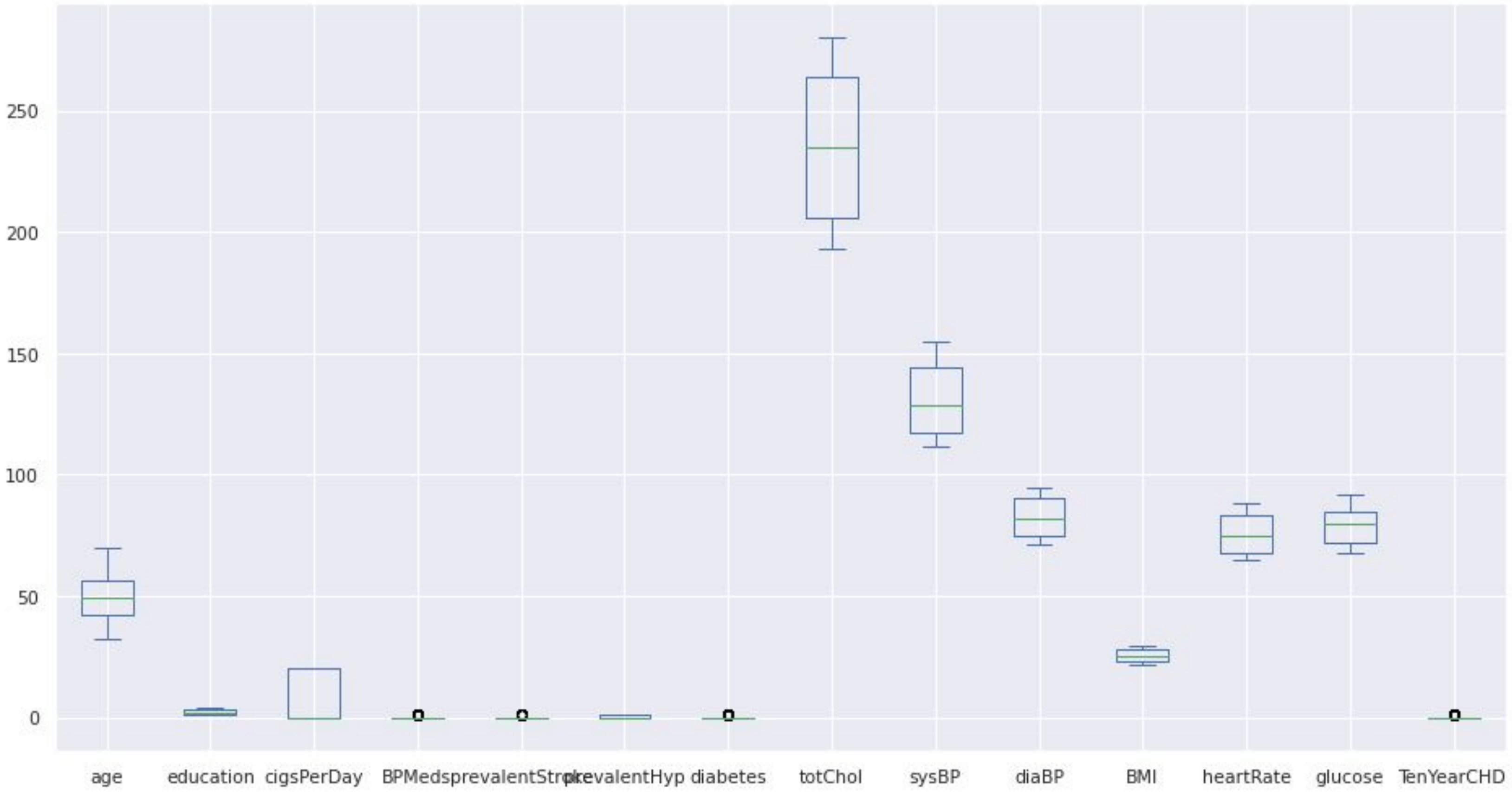
- Slightly more males are suffering from CHD than females .
- The people who has high BMI are at risk of CHD .
- The people with hypertension are at high risk of CHD
- The percentage of people who have CHD is almost equal between smokers and non-smokers .
- The uneducated people or the people with basic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

Treatment of Missing Values and Outliers



By the above boxplot we will which features having outliers. cigsPerDay,totChol,sysBP,diaBP,BMI,heartRate,glucose. these features having outliers..

After Outlier Treatment



Feature Engineering

AI

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

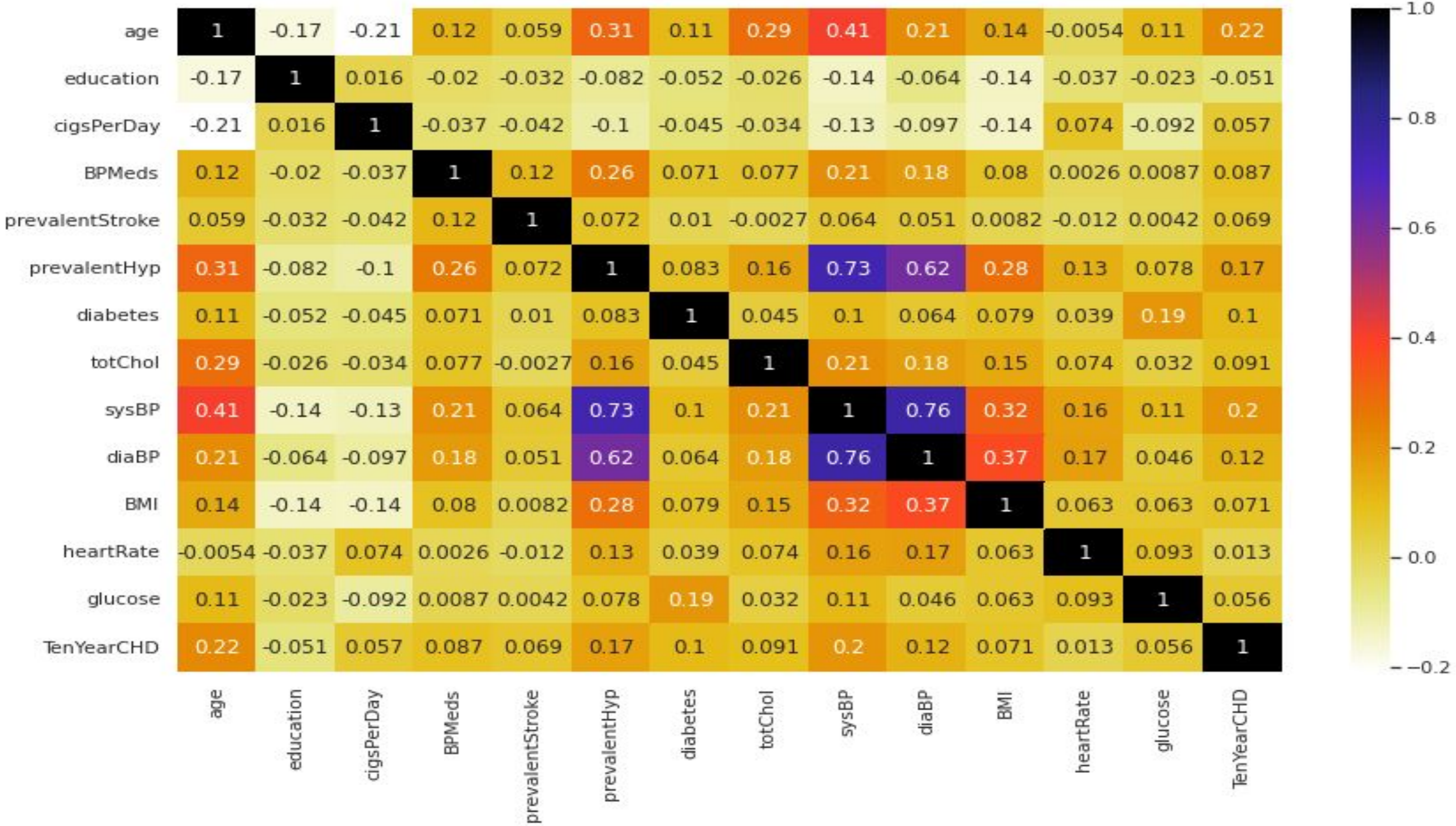
- Feature Engineering consists of various process :

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

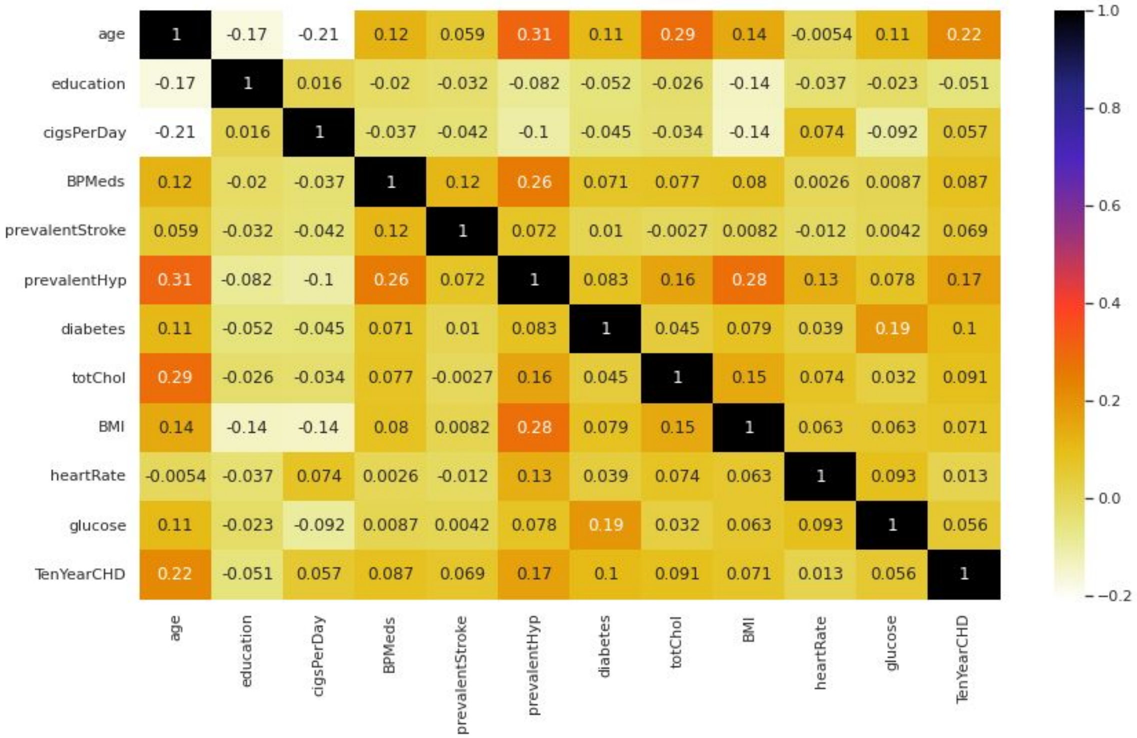
2. **Transformations:** Feature transformation is simply a function that transforms features from one representation to another(Normal distribution). We have used Box-cox and log transformation to convert columns to Normal distribution.

3. **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

Multicollinearity

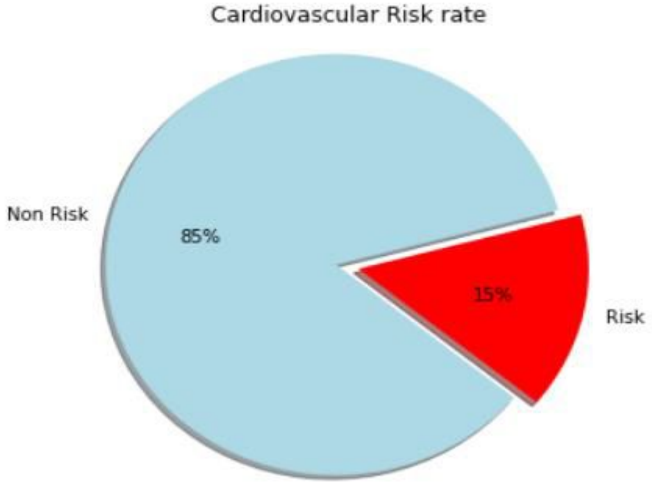


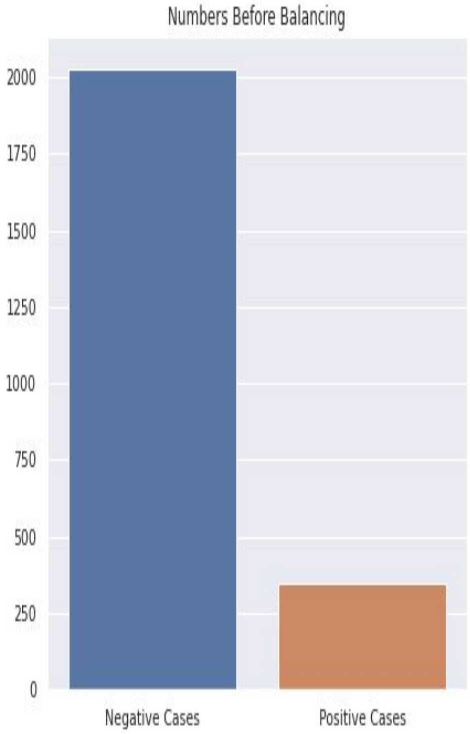
Multicollinearity

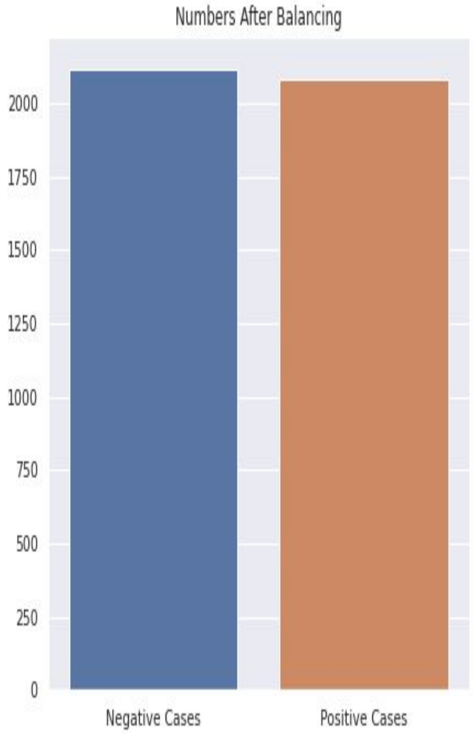


As we see in heatmap sysBp and diaBp is highly correlated up to 76 percentage so we have removed that column.

Data Balancing







We using **smote technique** for over sampling we have train set of size 3960 with 3960 samples of each of the class. Our dataset is now ready for training.

Model development

AI

Now we start building models for our classification problem. We have used following ML Algorithms:

(1)Logistic regression

(2)Support Vector Machine (SVM)

(3)K nearest neighbour (KNN)

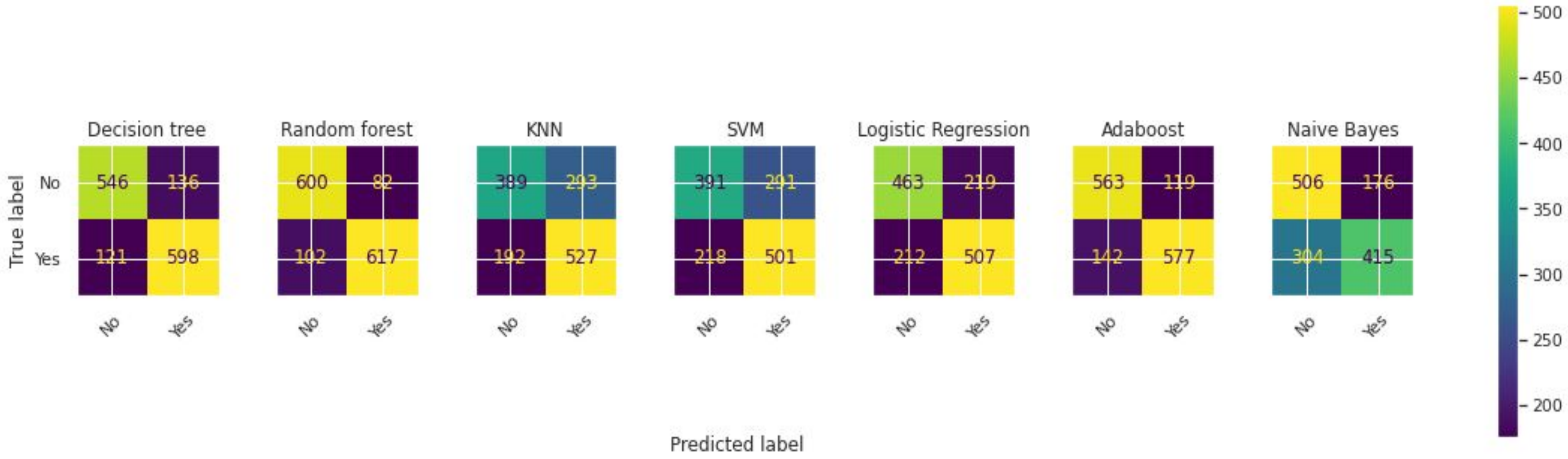
(4)Naive Bayes

(5)Decision Tree

(6)Adaboost

(7)Random Forest

Confusion Matrix



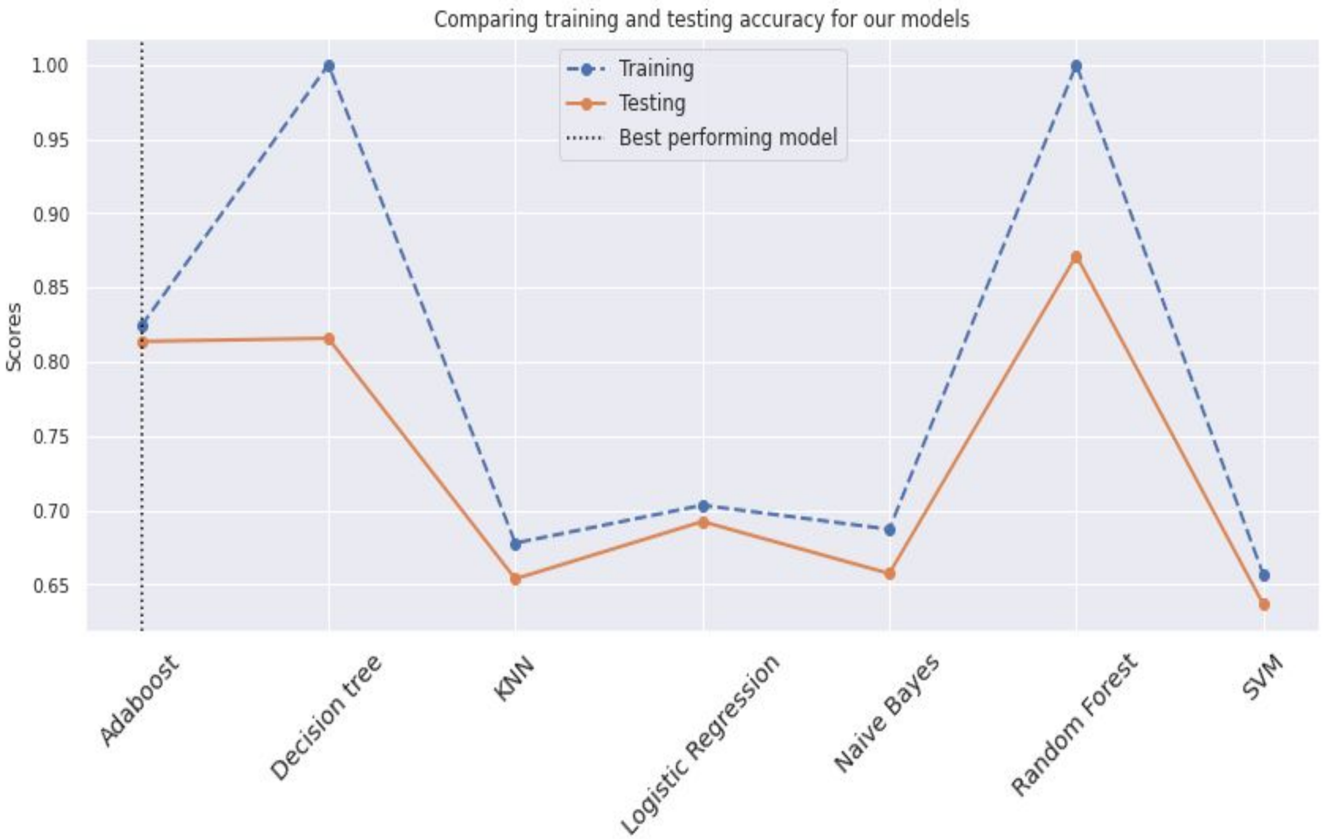
	Name	True_positive	False_positive	False_negative	True_negative	Correct_prediction	Wrong_prediction
0	Decision tree	553	130	104	619	1172	234
1	Random forest	625	58	87	636	1261	145
2	KNN	425	258	200	523	948	458
3	SVM	412	271	222	501	913	493
4	Logistic Regression	452	231	190	533	985	421
5	Adaboost	583	100	136	587	1170	236
6	Naive Bayes	499	184	260	463	962	444

Results & Model Performance

AI

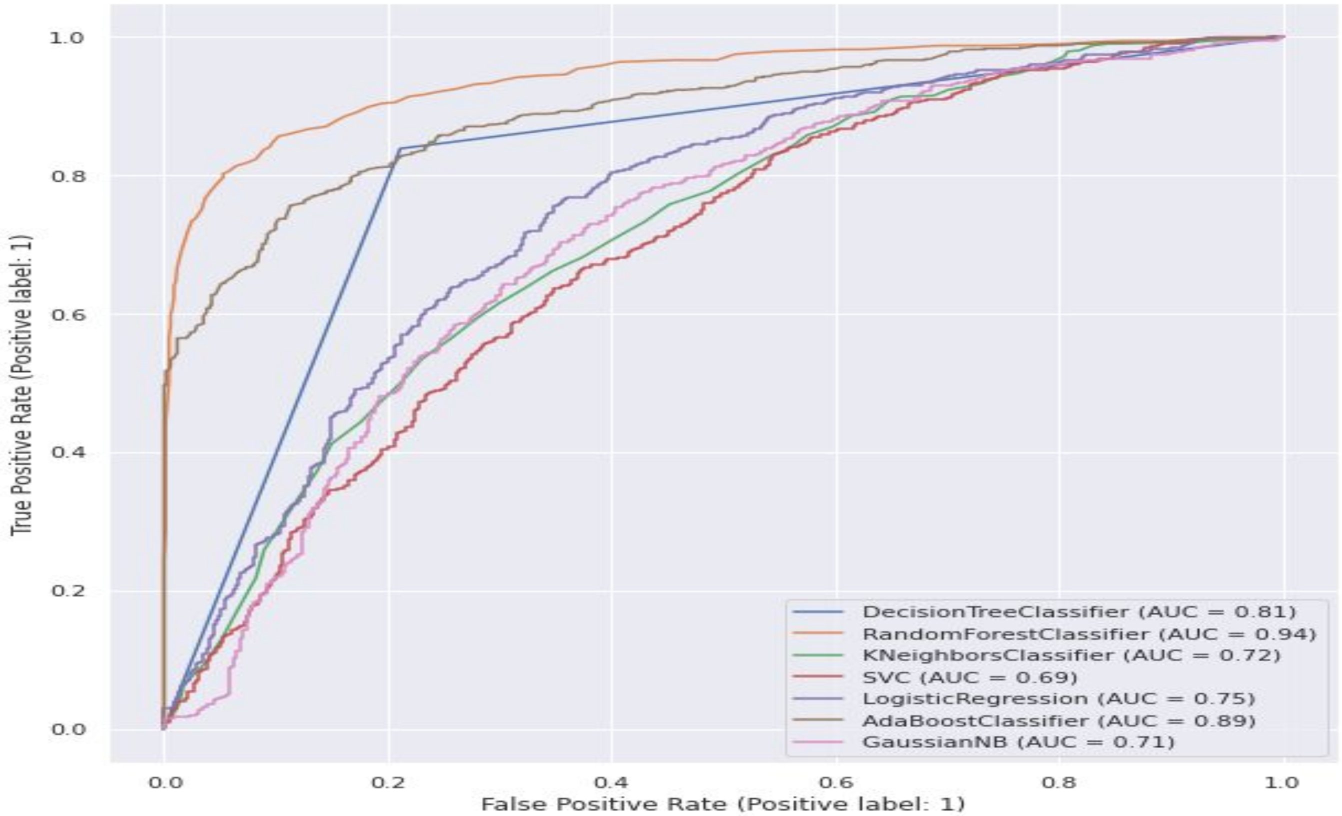
	Name	Train_accuracy	Test_accuracy	Precision	Recall	F1_Score
0	Adaboost	0.830963	0.832148	0.853587	0.810848	0.831669
1	Decision tree	1	0.825036	0.802343	0.831563	0.816692
2	KNN	0.671408	0.674253	0.622255	0.68	0.649847
3	Logistic Regression	0.700569	0.700569	0.661786	0.70405	0.682264
4	Naive Bayes	0.685633	0.684211	0.7306	0.657444	0.692094
5	Random forest	1	0.897582	0.915081	0.879044	0.8967
6	SVM	0.655524	0.64936	0.603221	0.649842	0.625664

Conclusions:



- We already had concluded that ada boost, Logistic Regression are performing well.
- Let's just try all other models with hyperparameter tuning and we will try to observe the performance of these models.

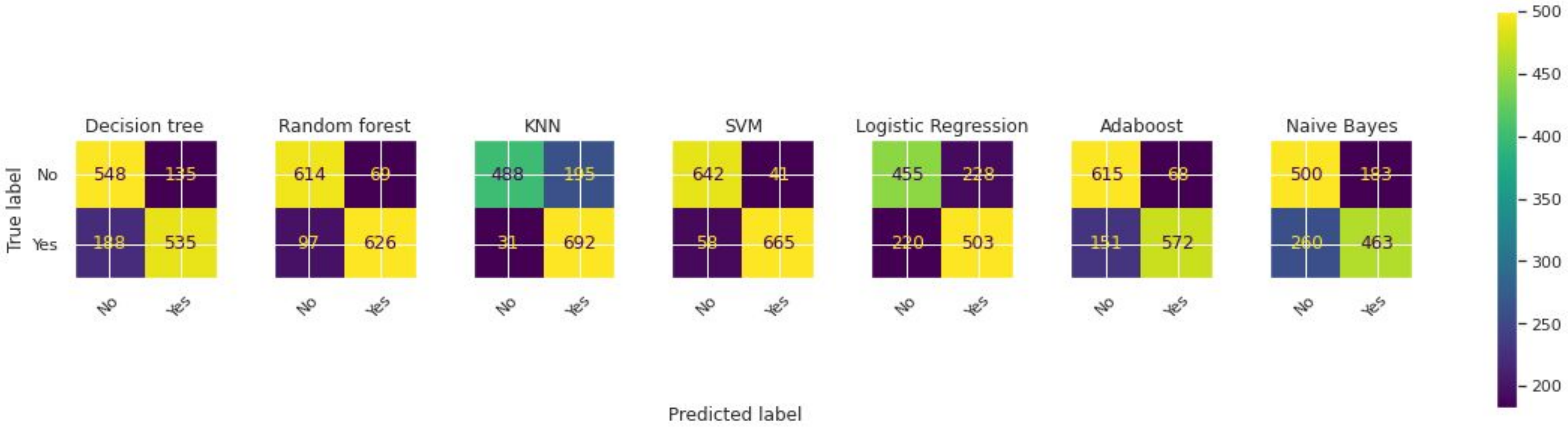
ROC Curves



1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)
DecisionTreeClassifier (AUC = 0.81)
RandomForestClassifier (AUC = 0.94)
KNeighborsClassifier (AUC = 0.72)
SVC (AUC = 0.69)
LogisticRegression (AUC = 0.75)
AdaBoostClassifier (AUC = 0.89)
GaussianNB (AUC = 0.71)

Confusion Metric after Hyperparameter Tuning for all models

AI



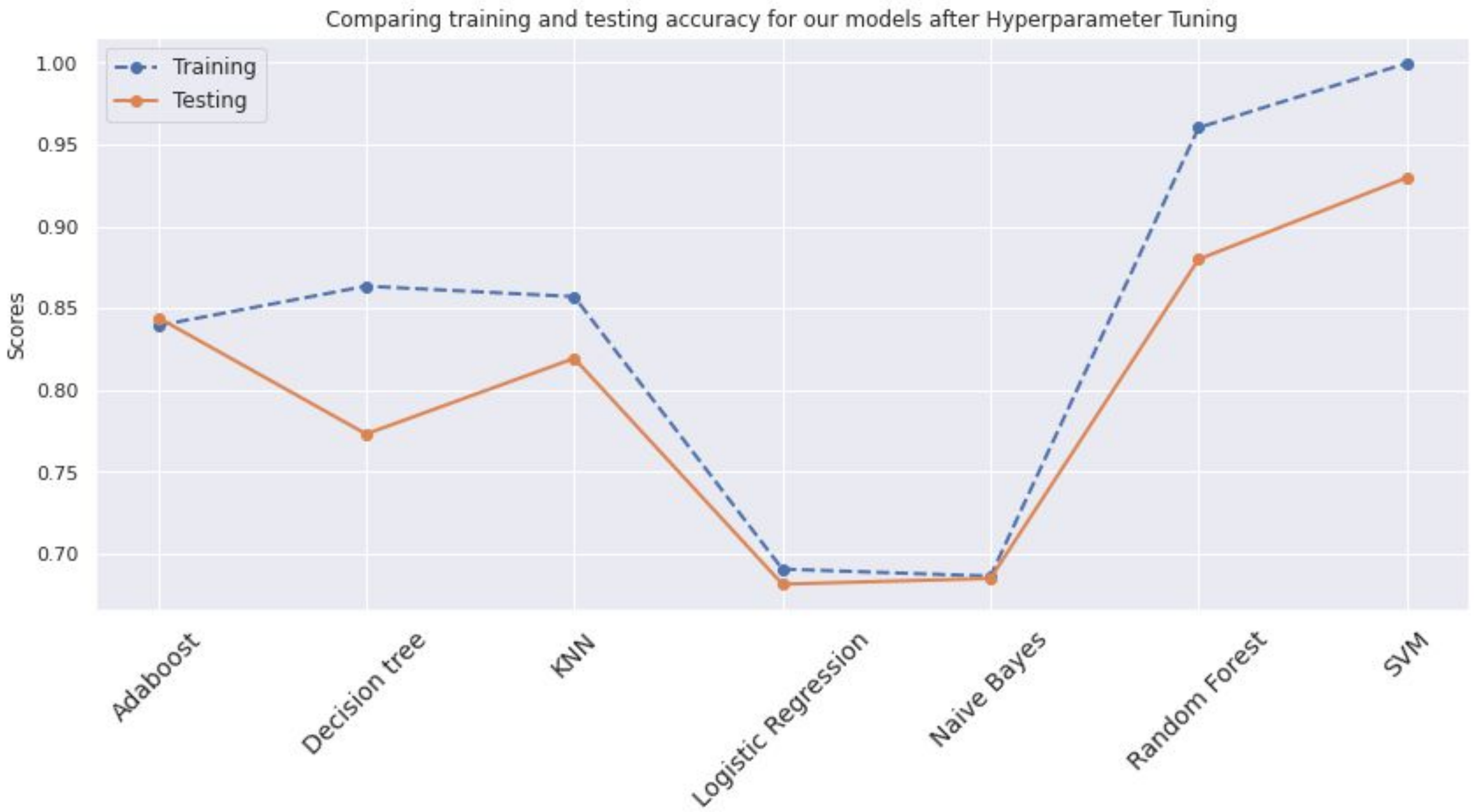
	Name	True_positive	False_positive	False_negative	True_negative	Correct_prediction	Wrong_prediction
0	Decision tree	567	116	179	544	1111	295
1	Random forest	614	69	98	625	1239	167
2	KNN	488	195	31	692	1180	226
3	SVM	642	41	58	665	1307	99
4	Logistic Regression	455	228	220	503	958	448
5	Adaboost	615	68	151	572	1187	219
6	Naive Bayes	500	183	260	463	963	443

Results & Model Performance after Hyperparameter Tuning

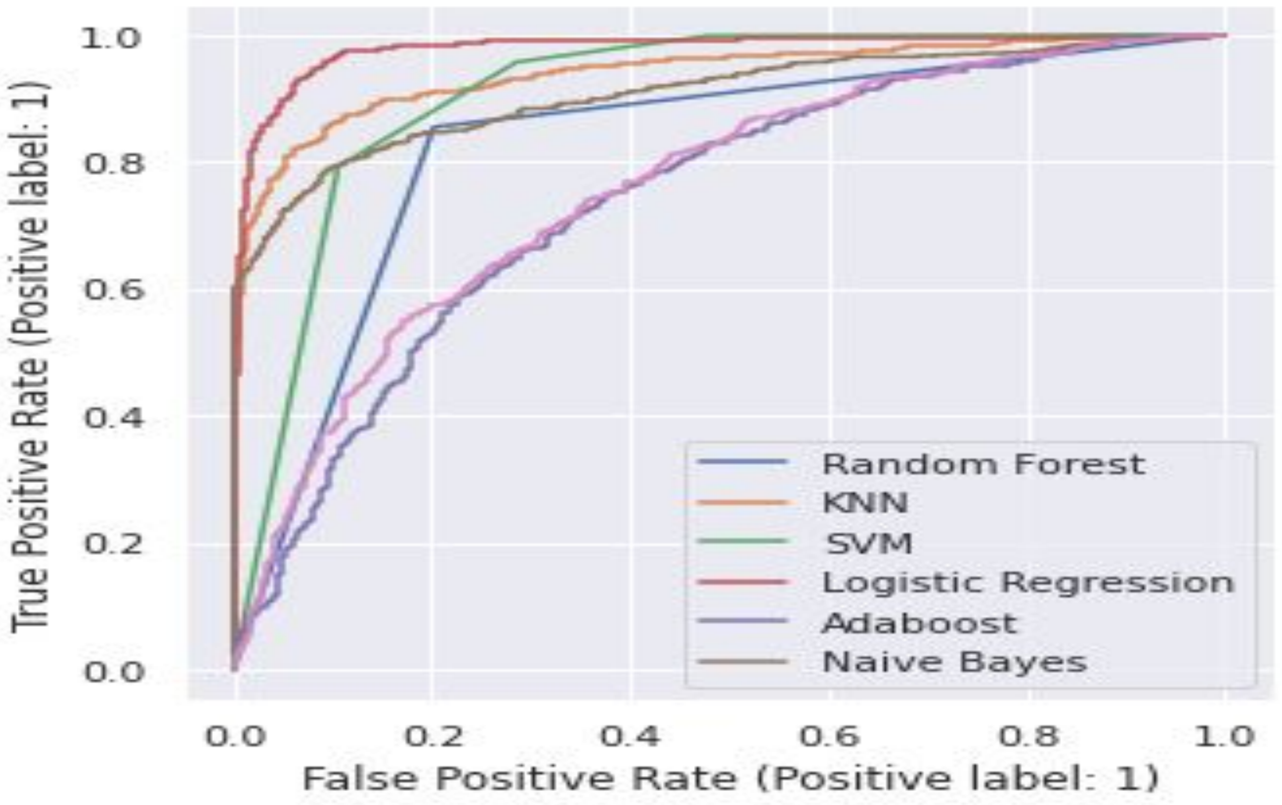
AI

	Name	Train_accuracy	Test_accuracy	Precision	Recall	F1_Score
0	Adaboost	0.830963	0.832148	0.853587	0.810848	0.831669
1	Adaboost after Hyperparameter Tuning	0.83926	0.844239	0.900439	0.802872	0.848861
2	Decision Tree	1	0.831437	0.79795	0.846273	0.821402
3	Decision Tree after Hyperparameter Tuning	0.88715	0.805121	0.809663	0.7934	0.801449
4	KNN	0.671408	0.674253	0.622255	0.68	0.649847
5	KNN after Hyperparameter tuning	0.857041	0.819346	0.692533	0.914894	0.788333
6	Logistic Regression	0.700569	0.700569	0.661786	0.70405	0.682264
7	Logistic Regression after Hyperparameter Tuning	0.690612	0.681366	0.666179	0.674074	0.670103
8	Naive Bayes	0.685633	0.684211	0.7306	0.657444	0.692094
9	Naive Bayes after Hyperparameter tuning	0.686344	0.684922	0.732064	0.657895	0.693001
10	Random Forest	1	0.894737	0.910688	0.877292	0.893678
11	Random Forest after Hyperparameter Tuning	0.952584	0.876956	0.888726	0.862216	0.87527
12	SVM	0.655524	0.64936	0.603221	0.649842	0.625664
13	SVM after Hyperparameter Tuning	0.999526	0.929587	0.939971	0.917143	0.928416

Conclusions:



ROC Curves after Hyperparameter Tuning for all models



Results & Conclusions:

- Even though the accuracy of some models were increased after hyper parameter tuning, adaboost, svm and Random forest are the best performing models.

- Between those if we compare tuned svm,Random forest and Adaboost with the accura cy and recall, tuned SVM, Adaboost are the best performance model for training data as well as test data.

- After Hyperparameter tuning also, we saw some models are facing overfitting (i.e., Random Forest and Decision Tree)

Challenges faced

AI

- The dependent variable has very few data labelled 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

- Machine learning models usually required high computation power

- We are doing hyper parameter tuning for all models so, it takes lot of time to run the Code.

References and links

References:

- Alma better
- Analytics Vidhya
- Kaggle
- Quora
- Stack overflow

Links for the code:

Github link: https://github.com/AruntejaLonka/Cardiovascular-Risk-Prediction

Google Drive link: https://colab.research.google.com/drive/1iRC-BREVXjjCwt0rOgzNRVEzzYr7I0fv?usp=sharing