Cardiovascular Risk Prediction

Capstone Project Detailed Report

By
Saurabh Aradwad

1. Abstract

A key component of preventive cardiology is identifying people who are at risk of cardiovascular disease (CVD). Clinical guidelines currently recommend risk prediction models that are based on a small number of predictors that perform poorly across all patient groups. Data-driven machine learning (ML) techniques may improve risk prediction performance by agnostically discovering novel risk predictors and learning the complex interactions between them. We investigated whether ML techniques based on a cutting-edge automated ML framework (AutoPrognosis) can predict a patient's 10-year risk of coronary heart disease (CHD).

2. Problem Statement

Because of their superiority in pattern recognition and classification when compared to other traditional statistical approaches, doctors and scientists alike have turned to machine learning (ML) techniques to develop screening tools.

In this project, we have used different Machine Learning techniques to predict whether a patient has a 10-year risk of developing coronary heart disease (CHD) on the Framingham, Massachusetts town dataset.

3. Introduction

Heart disease is a major health problem in India. According to the World Health Organization (WHO), cardiovascular diseases are the leading cause of death in India, accounting for more than 2.5 million deaths each year. The most common type of heart disease in India is coronary artery disease, which is caused by the accumulation of plaque in the arteries that supply blood to the heart. Other types of heart disease prevalent in India include heart failure, valvular heart disease, and rheumatic heart disease. Risk factors for heart disease in India include high blood pressure, high cholesterol, diabetes, tobacco use, and unhealthy diet.
This classification case study will help us to understand important factors which contribute to heart related diseases. Supervised machine learning's classification model we have created for the prediction of whether the patient has a 10-year risk of developing coronary heart disease.

4.Data Summary

The data set is publicly available on the Kaggle website, and it is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The classification goal is to predict whether the patient has 10-year risk of future coronary heart disease (CHD). The data set provides the patients' information. It includes over 4,000 records and 15 attributes. Each attribute is a potential risk factor. There are both demographic, behavioural and medical risk factors.

Attributes

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)
- **is_smoking** — whether or not the patient is a current smoker ("YES" or "NO")
- **cigsPerDay** — 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)

- **BPMeds** — whether or not the patient was on blood pressure medication (Nominal)
- **prevalentStroke** — whether or not the patient had previously had a stroke (Nominal)
- **prevalentHyp** — whether or not the patient was hypertensive (Nominal)
- **diabetes** — whether or not the patient had diabetes (Nominal)

Medical(current)

- **totChol** total cholesterol level (Continuous)
- **sysBP** systolic blood pressure (Continuous)
- **diaBP** 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 a large number of possible values.)
- **Glucose** glucose level (Continuous)

Predict variable (desired target)

- 10 year risk of developing coronary heart disease (CHD) - (binary: “1”, means “There is a risk”, “0” means “There is no risk”).

5. Missing Data Analysis and Data Cleaning

Missing data, also known as missing values, refers to data that is not complete or not available for analysis. In a dataset, missing data can occur for a variety of reasons, such as data that was not collected, data that was not entered correctly, or data that was lost due to a technical issue. Handling missing data is an important task in data analysis because the presence of missing values can affect the accuracy and reliability of the results.

There are several strategies for dealing with missing data, including dropping the rows or columns with missing values, imputing the missing values with statistical methods, or using machine learning techniques to predict the missing values. It is important to carefully consider the best approach for handling missing data in a given dataset, as the chosen strategy can have a significant impact on the results of the analysis.

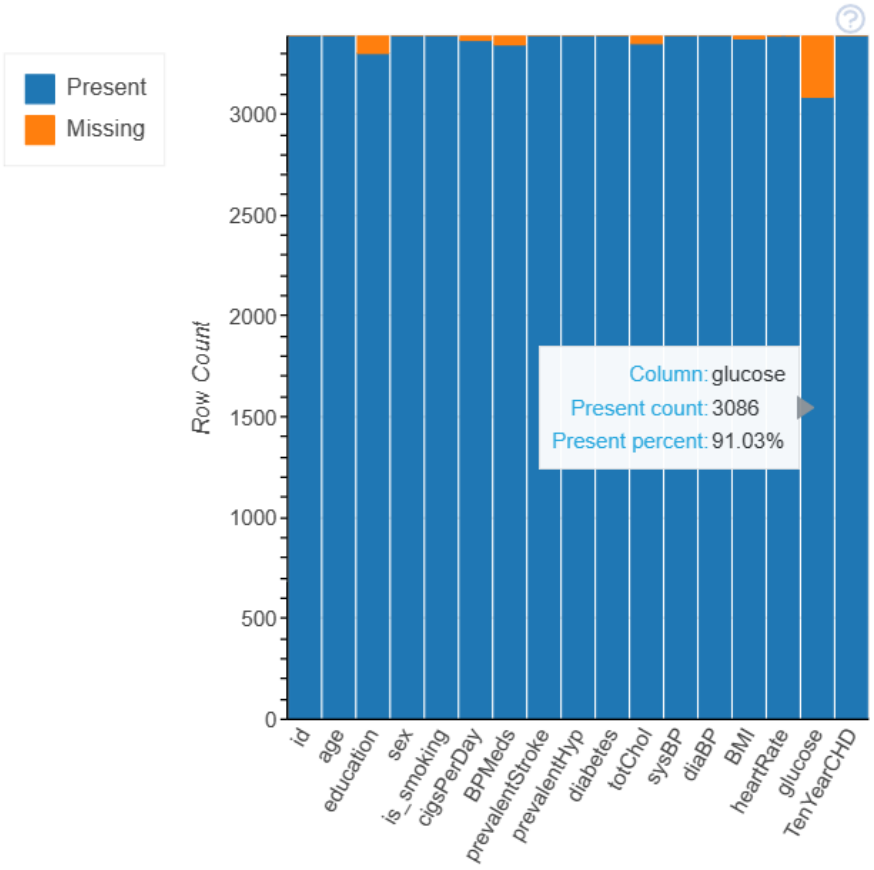
For this given dataset we considered total data volume and calculated the total percentage of missing data. If the data volume is less than 15% of total data then we can exclude this data for model creation.

Below we have shown missing statistics and visualisation for missing data. This is created with the DataPrep library.

Stats | Bar Chart | Spectrum | Heat Map | Dendrogram

Missing Statistics

Missing Cells	510
Missing Cells (%)	0.9%
Missing Columns	7
Missing Rows	463
Avg Missing Cells per Column	30.0
Avg Missing Cells per Row	0.15



	Total	Percentage
glucose	304	8.967552
education	87	2.566372
BPMeds	44	1.297935
totChol	38	1.120944
cigsPerDay	22	0.648968
BMI	14	0.412979
heartRate	1	0.029499

This table shows the percentage of missing data from each column. The treatment for missing data was as follows

- At 8.97%, the blood glucose entry has the highest percentage of missing data. The other features have very few missing entries.
- Since the missing entries account for only 11% of the total data, we can exclude these entries without losing most of the data.

6. Exploratory Data Analysis

Exploratory data analysis (EDA) is an approach to analysing and understanding data that involves developing insights through a process of visualising, summarising, and reasoning about the data. It is a way to get to know the data you are working with, uncover patterns and relationships, and identify any potential problems or limitations in the data.

7.1 Integer Treatment

Before we go ahead, an important step to do is to convert our string feature into an integer.
In sex feature M will be converted to 1 and F will be converted to 0.
In is_smoking feature YES will be converted to 1 and NO will be converted to 0.

7.2 Dataprep Report

The goal of create_report is to generate profile reports from a pandas DataFrame. create_report utilises the functionalities and formats the plots from dataprep. It provides the following information:

1. **Overview**: detect the types of columns in a dataframe
2. **Variables**: variable type, unique values, distinct count, missing values
3. **Quantile statistics** like minimum value, Q1, median, Q3, maximum, range, interquartile range
4. **Descriptive statistics** like mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness
5. **Text analysis** for length, sample and letter
6. **Correlations**: highlighting of highly correlated variables, Spearman, Pearson and Kendall matrices
7. **Missing Values**: bar chart, heatmap and spectrum of missing values

In the following, we break down the report into different sections to demonstrate each part of the report.

To view report generated click here

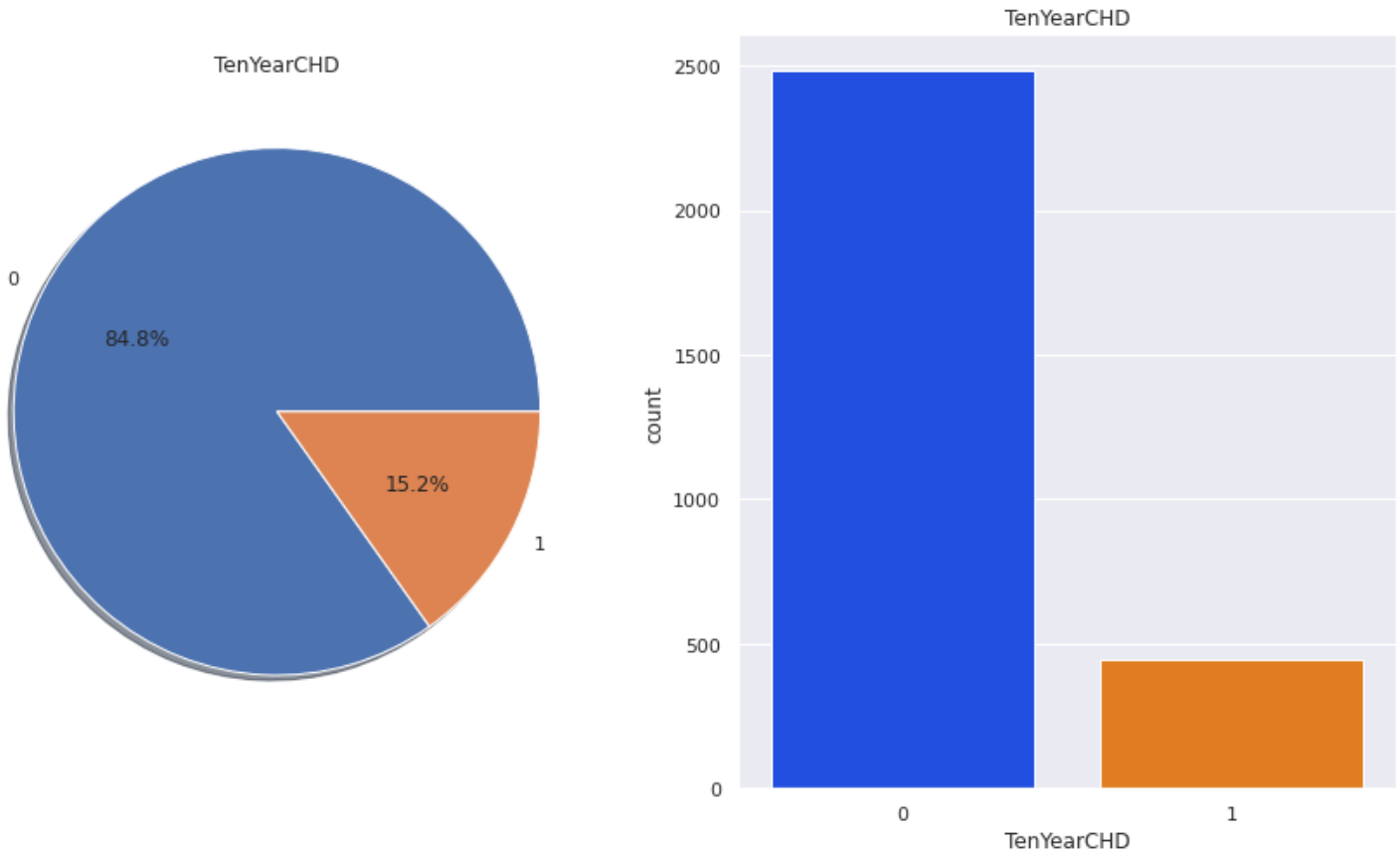
7.3 Feature Analysis

Feature analysis is a process of identifying and understanding the relationships between the variables or features in a dataset. It is often used as a way to select a subset of relevant features for building predictive models or to understand the importance of different features in the data.

Our **target variable** is TenYearCHD

There are 2483 patients without heart disease and 444 patients with the disease.

- 1 indicates person have risk of coronary heart disease
- 0 indicates person do not have risk of coronary heart disease

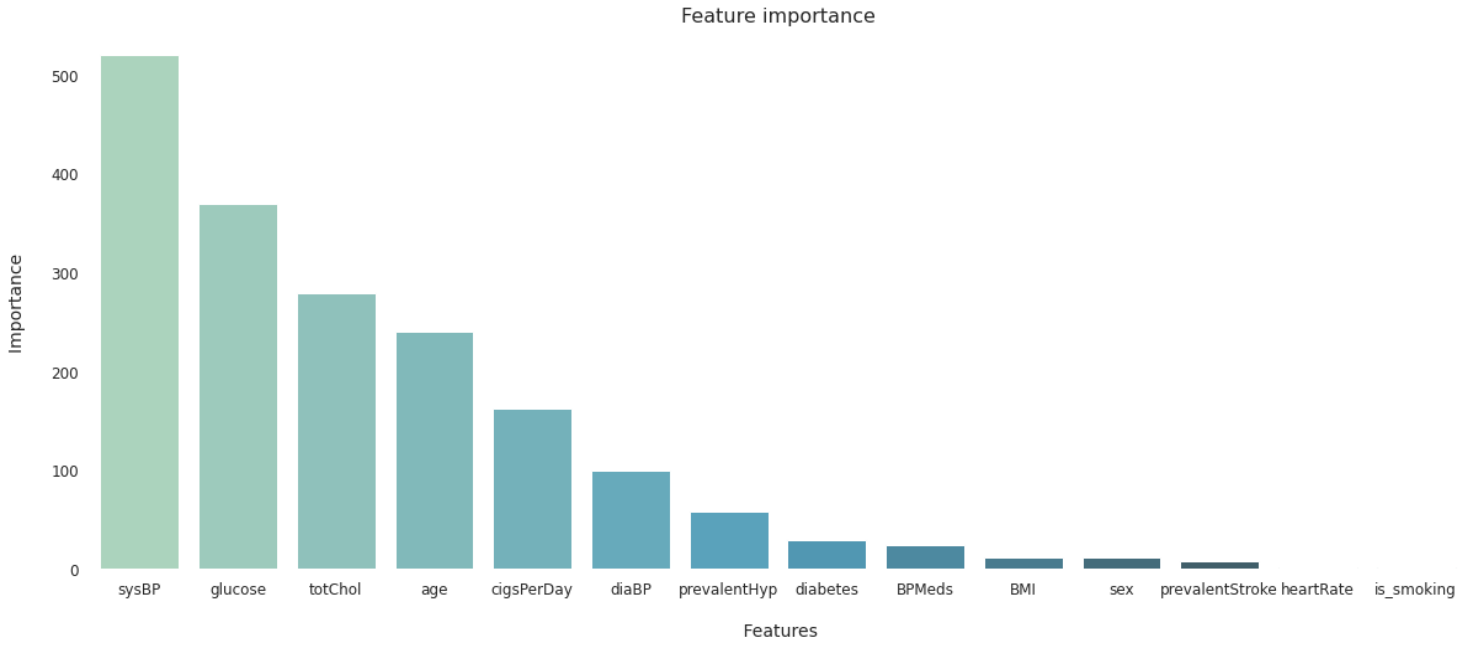


We can see above that we have an imbalanced data set as the number of people without the disease greatly exceeds the number of people with the disease.

15.2 % out of total people have **Cardiovascular Risk**.
84.8 % of total people are **risk free**.

7.4 Feature Selection

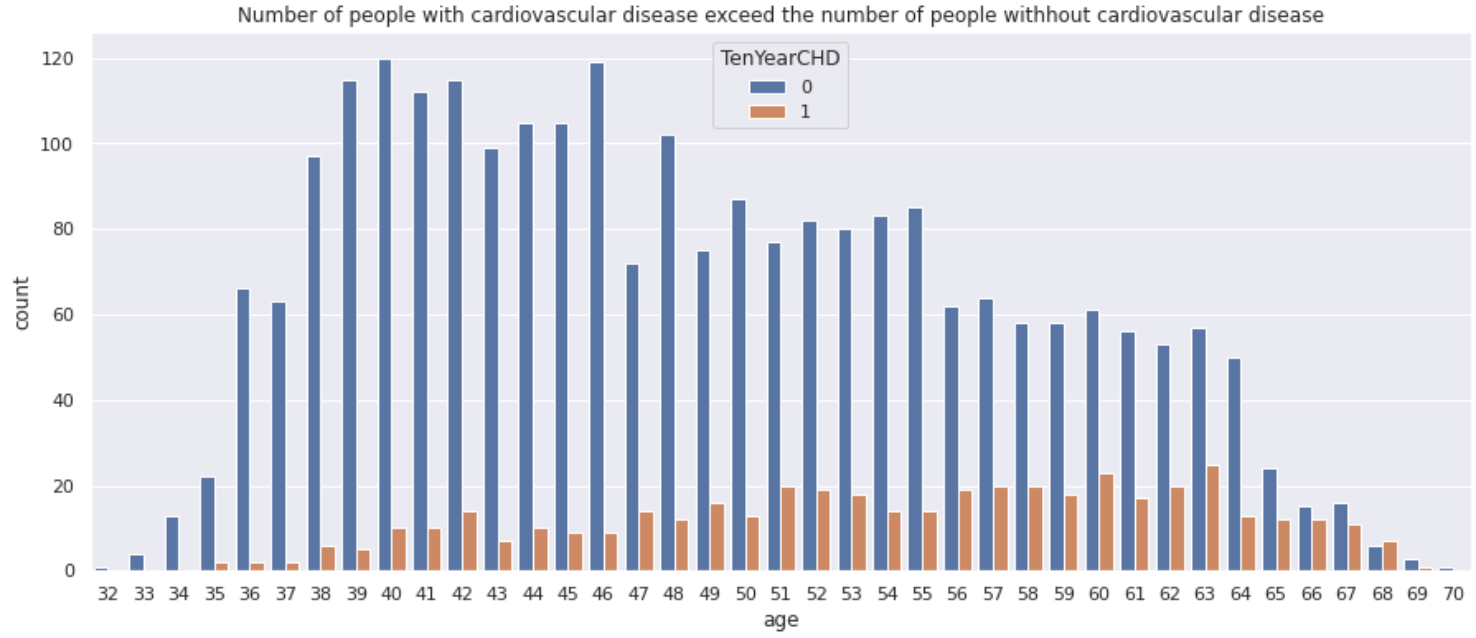
The classes in the sklearn.feature_selection module can be used for feature selection/dimensionality reduction on sample sets, either to improve estimators' accuracy scores or to boost their performance on very high-dimensional datasets.



From above visualisation we can consider features such as **sysBP**, **glucose**, **totChol**, **age**, **cigsPerDay**, **diaBP** who have very scores in model creation.

7.5 Attack on Target Variable

Detailed analysis of target variable is important so here we have analysed TenYearCHD. The following plot shows age vs TenYearCHD

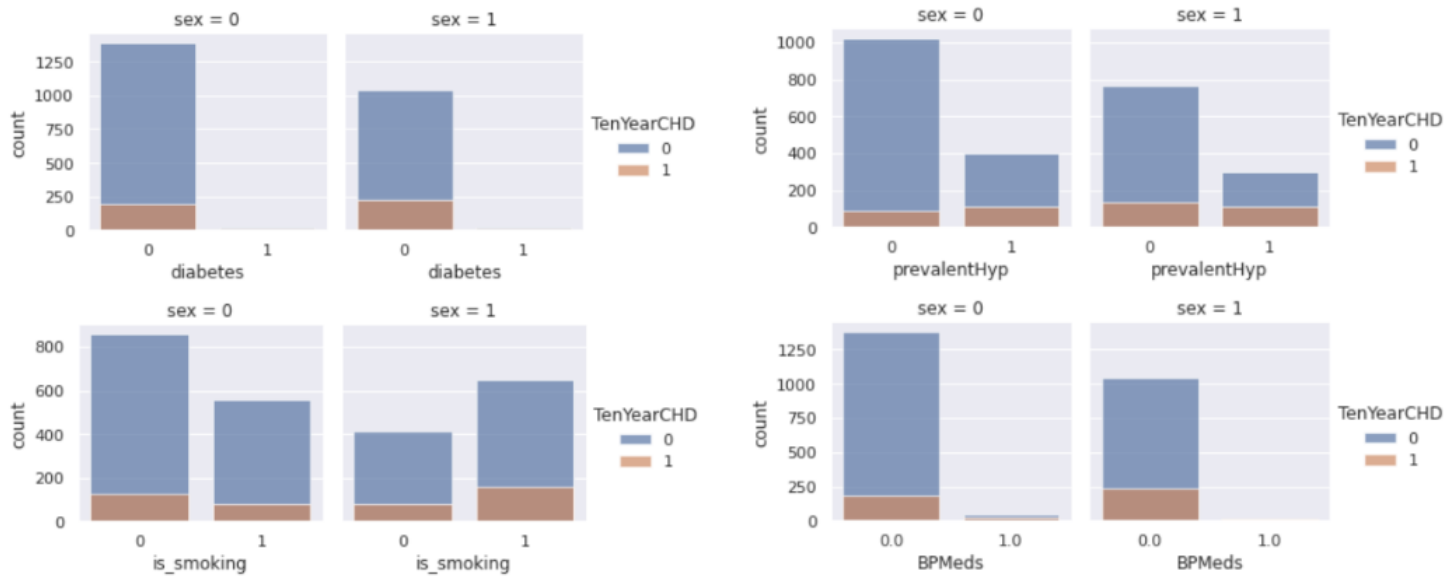


As we can see in the above plot The people with the highest risk of developing heart disease are between the ages of 51 and 63.

Because the number of sick people generally increases with age.

Categorical Variable Comparison

Plotting categorical variables with respect to target variable.



From the above categorical variables comparison plot we can conclude that,

Slightly more males are suffering from Cardiovascular heart disease than females.

- The number of people who have Cardiovascular heart disease is almost equal between smokers and non-smokers.
- The percentage of people who have Cardiovascular heart disease is higher among the diabetic patients and also those patients with prevalent hypertension have more risk of Cardiovascular heart disease compared to those who don't have hypertensive problems.
- The percentage of people who are on medication of blood pressure have more risk of Cardiovascular heart disease compared to those who are not on medication.

7. Outlier Treatment

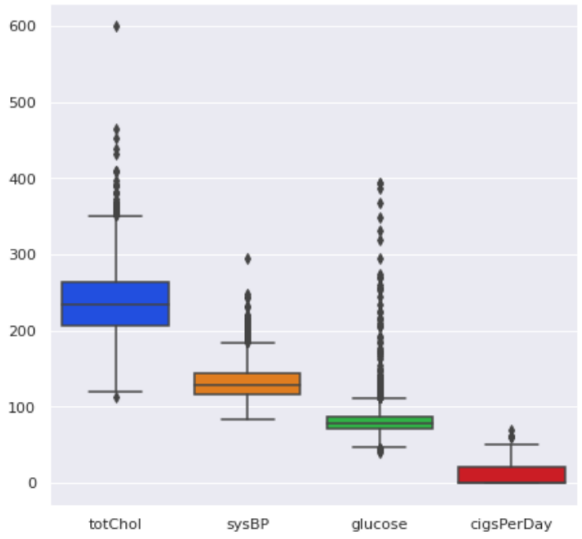
An outlier is a data point that is significantly different from the other data points in a dataset. Outliers can be caused by a variety of factors, including errors in data collection or measurement, variations in the underlying process that generated the data, or the presence of rare events.

Our approach for outlier treatment is as follows:

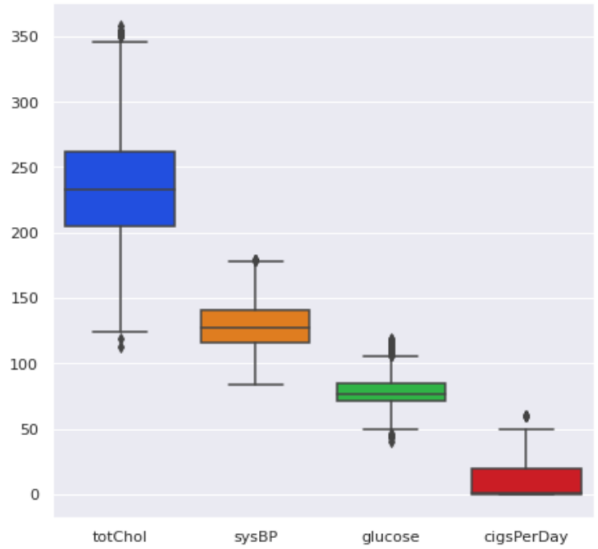
- Step 1: Create a boxplot for the entire dataframe and identify which columns have maximum outliers.
- Step 2: Create separate data frame consisting columns which have maximum outliers.

- Step 3: Remove outliers from original data frame with the help of boxplot and revalidate with boxplot.

  Following Fig shows boxplots before and after outlier treatment



Before Outlier Treatment



After Outlier Treatment

8. Correlation analysis

Correlation analysis is a statistical method used to examine the relationship between two variables. It allows you to determine whether there is a statistical relationship between the two variables and, if so, the strength and direction of that relationship.

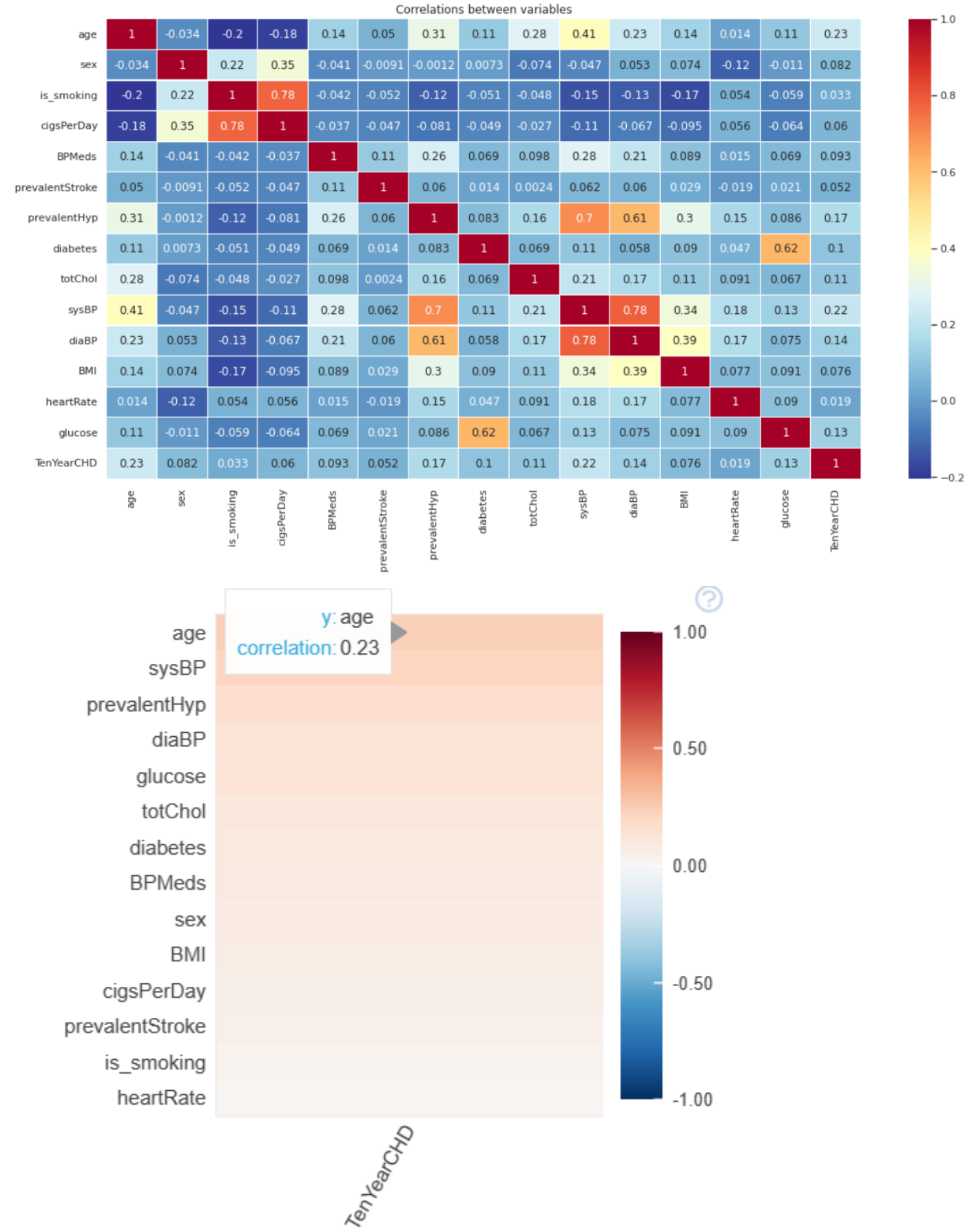
The result of a correlation analysis is a correlation coefficient, which can range from -1 to 1. A coefficient of -1 indicates a strong negative correlation, meaning that as the value of one variable increases, the value of the other variable decreases. A coefficient of 1 indicates a strong positive correlation, meaning that as the value of one variable increases, the value of the other variable also increases. A coefficient of 0 indicates that there is no relationship between the two variables.

For correlation analysis the best ways are to create heatmaps. We have created a heatmap with the help of seaborn heatmap and again revalidated this with the Datapreps correlation heatmap for a single column.

Some insights from Correlation plot are as follows:

- Max correlation obtained is : 0.23 for age column w.r.t TenYearCHD.
- Minimum Correlation obtained is : 0.02 for heart rate column w.r.t. TenYearCHD.

Following plots are obtained for correlation analysis



9. Feature Engineering

Feature engineering is the process of extracting useful features from raw data and creating new features that can help improve the performance of machine learning models. It is a critical step in the process of building machine learning models, as the quality of the features used as input to the model can have a significant impact on its performance.

Feature engineering involves a combination of domain knowledge and statistical techniques to identify and create relevant features from the raw data. It can include a wide range of activities, such as:

- Selecting relevant features from the raw data: This can involve identifying the most important features to include in the model based on domain knowledge or using statistical techniques to evaluate the importance of different features.
- Transforming or scaling features: This can involve normalising the scale of features, one-hot encoding categorical features, or using techniques such as binning to convert numeric features into categorical features.
- Creating new features: This can involve combining or aggregating existing features to create new ones, or using domain knowledge to create new features based on the problem at hand.

Overall, the goal of feature engineering is to create a set of features that will be useful for building a machine learning model that can accurately predict the output variable given the input data.

10.1 Selection of Top Features

Tree-based: SelectFromModel

SelectFromModel is an Embedded method. Embedded methods use algorithms that have built-in feature selection methods.

Here, We have used RandomForest() to select features based on feature importance. We calculate feature importance using node impurities in each decision tree.

In Random forest, the final feature importance is the average of all decision tree feature importance.

The top features are as follows:

Age, Total cholesterol, Systolic blood pressure, Diastolic blood pressure, BMI, Heart rate, Blood glucose

10.2 Pairplot with Top Features

A pairplot is a plot that allows you to visualise the relationships between different pairs of variables in a dataset. It is a matrix of plots, where each plot in the matrix represents a pair of variables and the scatterplot of their values.

Pairplots are useful for exploring the relationships between variables in a dataset and can be especially useful for identifying trends and patterns in the data. They can also be useful for identifying potential outliers in the data.

To see pairplot for our dataset click here

10. Preparing Dataset for Modeling

11.1 SMOT

Since our dataset is imbalanced i.e for every positive case there are about 5-6 negative cases. We may end up with a classifier that is biased to the negative cases. The classifier may have a high accuracy but poor precision and recall.

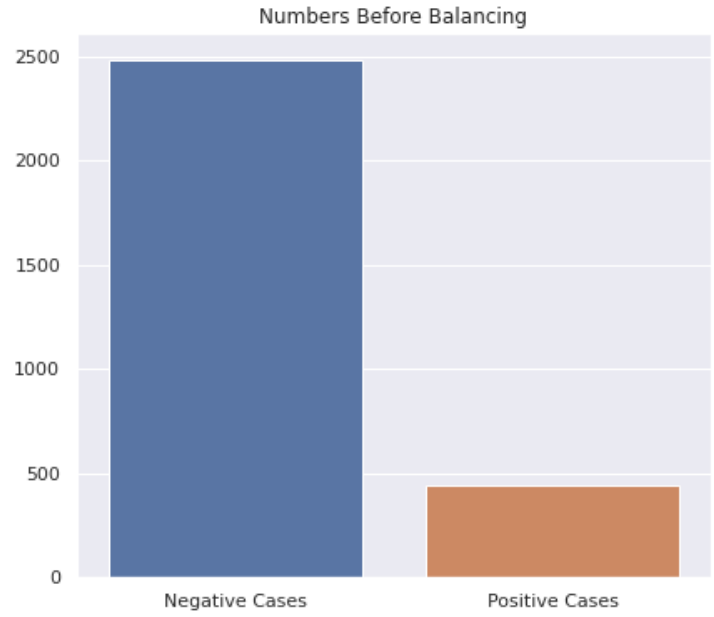
To handle this problem we will balance the dataset using the **Synthetic Minority Oversampling Technique** (SMOTE).

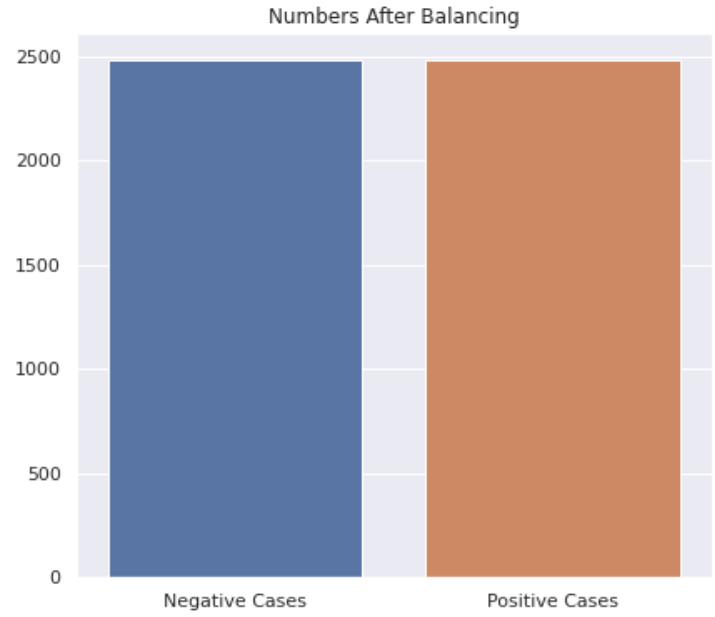
SMOTE works by randomly picking a point from the minority class and computing the k-nearest neighbours for this point. The synthetic points are then added between the chosen point and its neighbours.

SMOTE algorithm works in 4 simple steps:

- Step 1: Choose a minority class as the input vector
- Step 2: Find its k nearest neighbours (k_neighbors is specified as an argument in the SMOTE() function)
- Step 3: Choose one of these neighbours and place a synthetic point anywhere on the line joining the point under consideration and its chosen neighbour
- Step 4: Repeat the steps until data is balanced.

This following plot shows visualisation before and after balancing the dataset with SMOT.





11.2 Split

Splitting data into a training set and a test set is a common practice in machine learning, and is a way to evaluate the performance of a model on unseen data. The idea is to use a portion of the data to train the model and then use the remaining portion to test the model's predictions.

We have splitted our new balanced data into 20% and 80% split for training and testing.

11.3 Metrics

We use evaluation metrics for comparing our model. For this classification problem we have use some metrics and they are as follows:

- **GridSearchCV** implements a “fit” and a “score” method. It also implements “score_samples”, “predict”, “predict_proba”, “decision_function”, “transform” and “inverse_transform” if they are implemented in the estimator used.The parameters of the estimator used to apply these methods are optimised by cross-validated grid-search over a parameter grid.
- **confusion_matrix** to evaluate the accuracy of a classification.
- **classification_report** builds a text report showing the main classification metrics.

For evaluation of classification model scores we have considered accuracy_score, f1_score, recall_score, prescision_score, roc_auc_score, roc_curve.

1. The accuracy score:
which is the ratio of the number of correct predictions to the total number of input samples. It measures the tendency of an algorithm to classify data correctly.

2. The F1 Score:
Which is defined as the weighted harmonic mean of the test's precision and recall. By using both precision and recall it gives a more realistic measure of a test's performance. (Precision, also called the positive predictive value, is the proportion of positive results that truly are positive. Recall, also called sensitivity, is the ability of a test to correctly identify positive results to get the true positive rate).

3. The Area under the ROC Curve (AUC):
Which provides an aggregate measure of performance across all possible classification thresholds. It gives the probability that the model ranks a random positive example more highly than a random negative example.

11. Classification Models

We have used the scikit-learn library for building classification models. The four algorithms that we will be using are:

1. Logistic Regression
2. Random Forest
3. XGBoost
4. Support Vector Machine

Here, we will be using the GridsearchCV search algorithm for above algorithms to identify best parameters.

Some reusable functions we created for model fitting, getting best parameters, model evaluation scores and error curve.

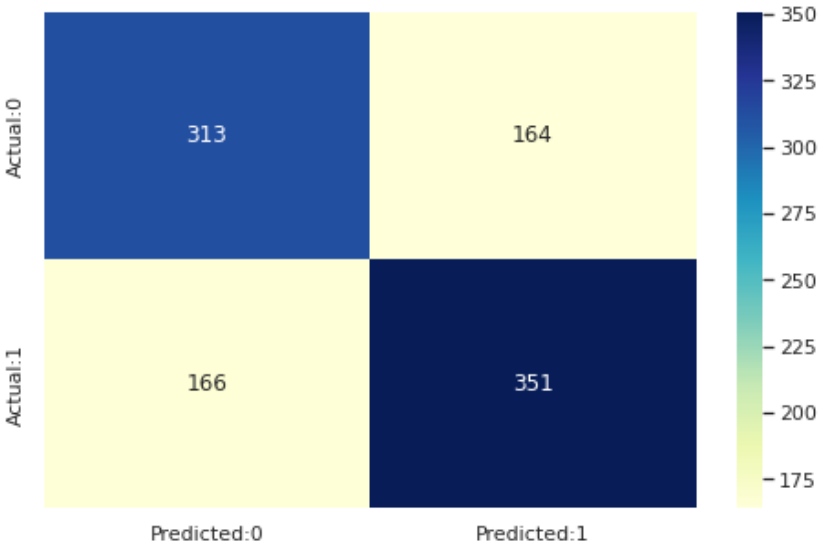
12.1 Logistic Regression

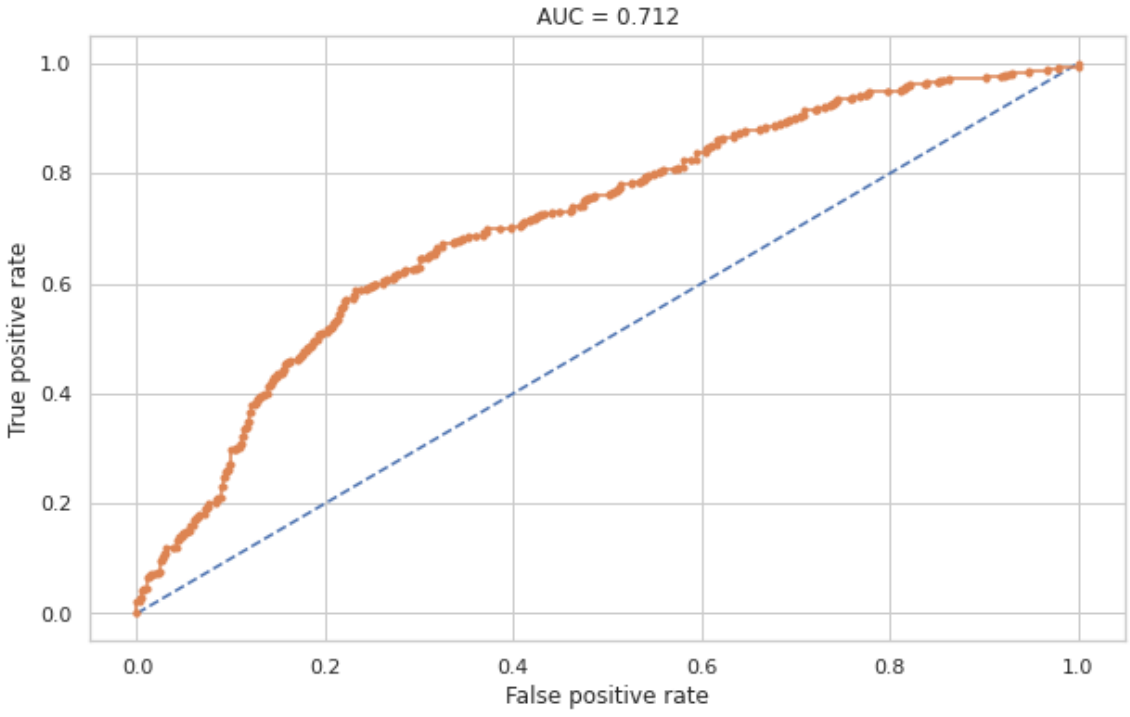
Logistic regression aims to measure the relationship between a categorical dependent variable and one or more independent variables (usually continuous) by plotting the dependent variables' probability scores.

Best parameters:
{'C': 10,
'class_weight': None,
'penalty': 'l2'}

Classification Report For This model is as follows

	precision	recall	f1-score	support
0	0.65	0.66	0.65	477
1	0.68	0.68	0.68	517
accuracy			0.67	994
macro avg	0.67	0.67	0.67	994
weighted avg	0.67	0.67	0.67	994





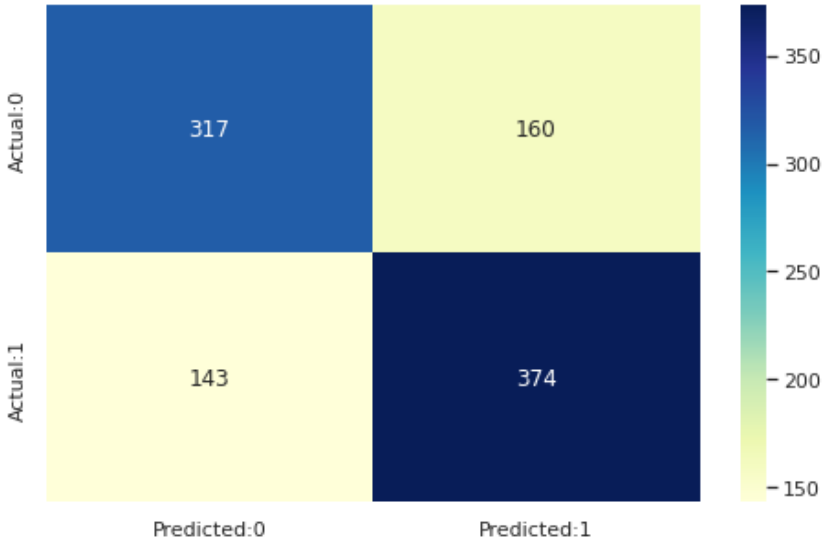
12.2 Random Forest

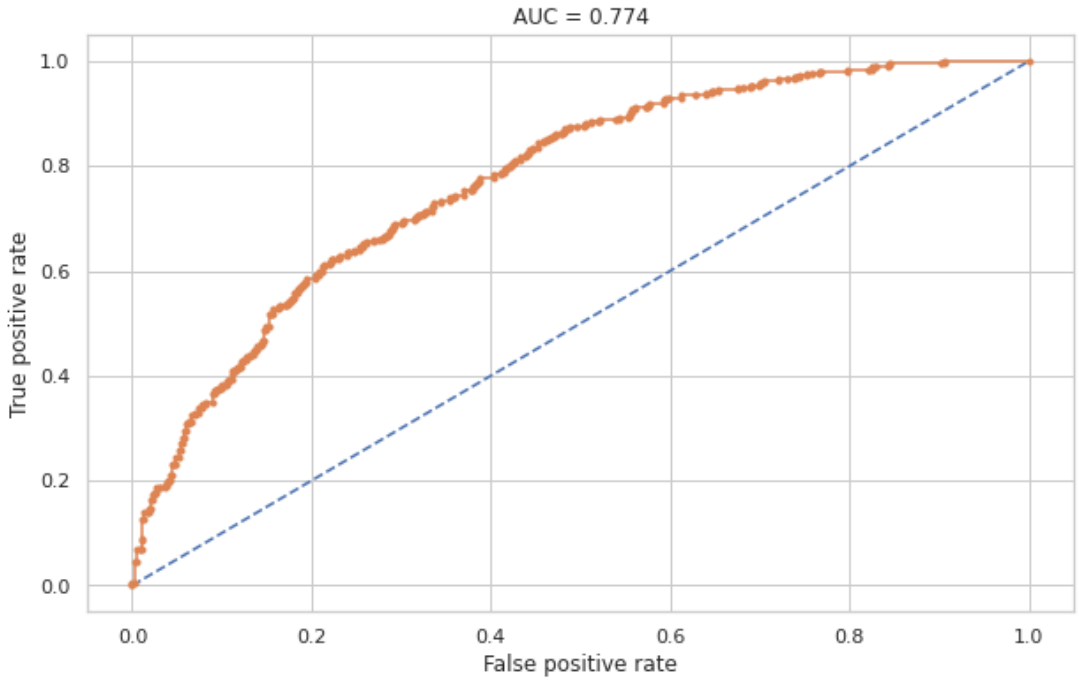
Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

Best parameters:
{'max_depth': 8,
'min_samples_leaf': 40,
'min_samples_split': 50,
'n_estimators': 80}

Classification Report For This model is as follows

	precision	recall	f1-score	support
0	0.69	0.66	0.68	477
1	0.70	0.72	0.71	517
accuracy			0.70	994
macro avg	0.69	0.69	0.69	994
weighted avg	0.69	0.70	0.69	994





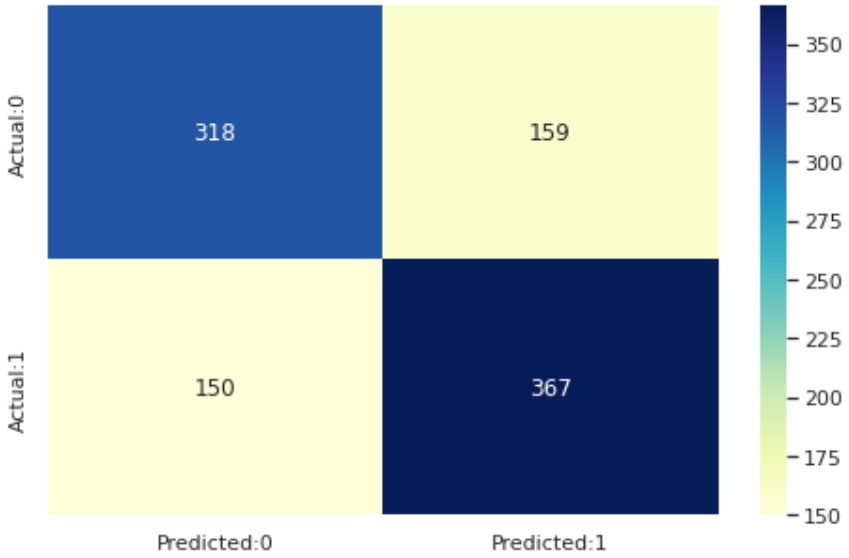
12.3 XG Boost

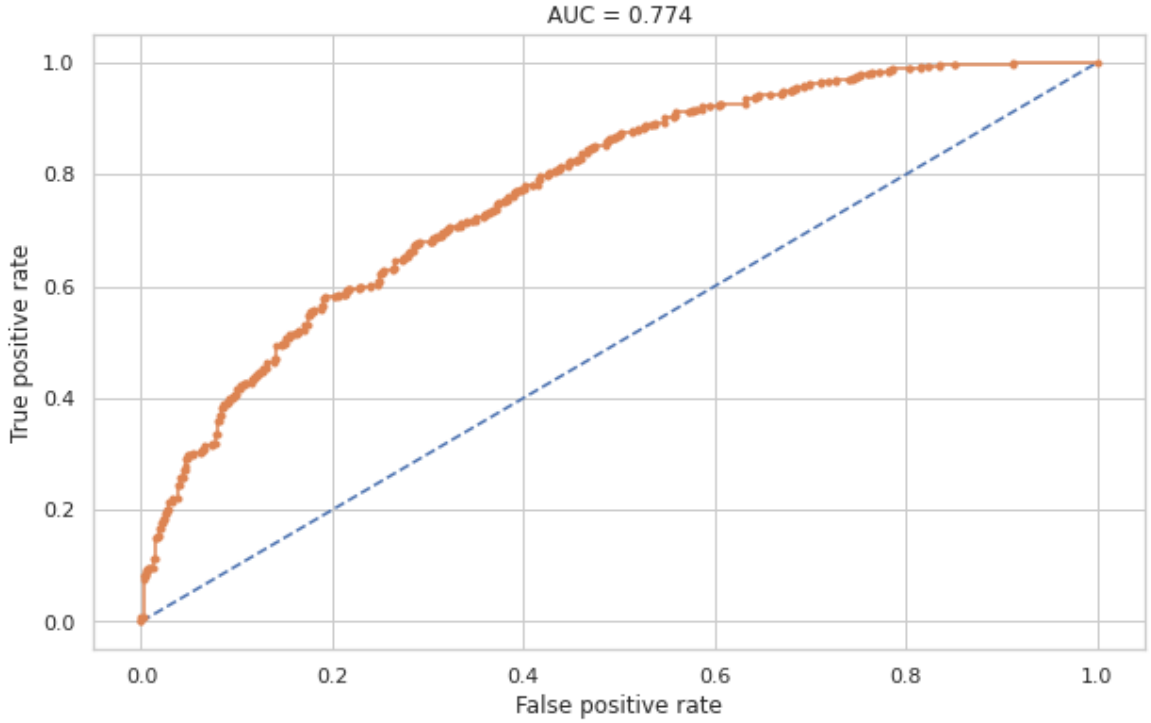
XGBoost stands for eXtreme Gradient Boosting. The name XGboost, though, actually refers to the engineering goal to push the limit of computational resources for boosted tree algorithms.

Best parameters:
{'learning_rate': 0.1,
'max_depth': 11,
'n_estimators': 200}

Classification Report For This model is as follows

	precision	recall	f1-score	support
0	0.68	0.67	0.67	477
1	0.70	0.71	0.70	517
accuracy			0.69	994
macro avg	0.69	0.69	0.69	994
weighted avg	0.69	0.69	0.69	994





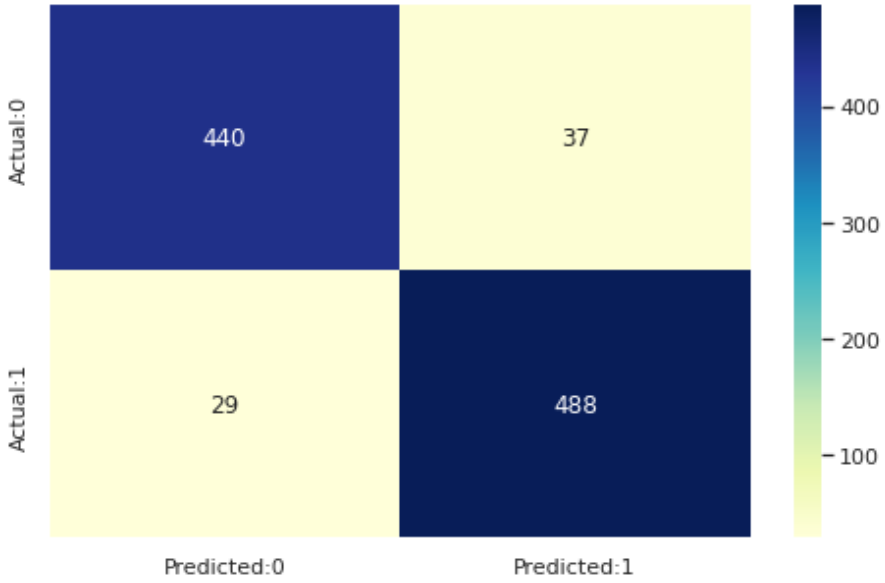
12.4 Support Vector Machine

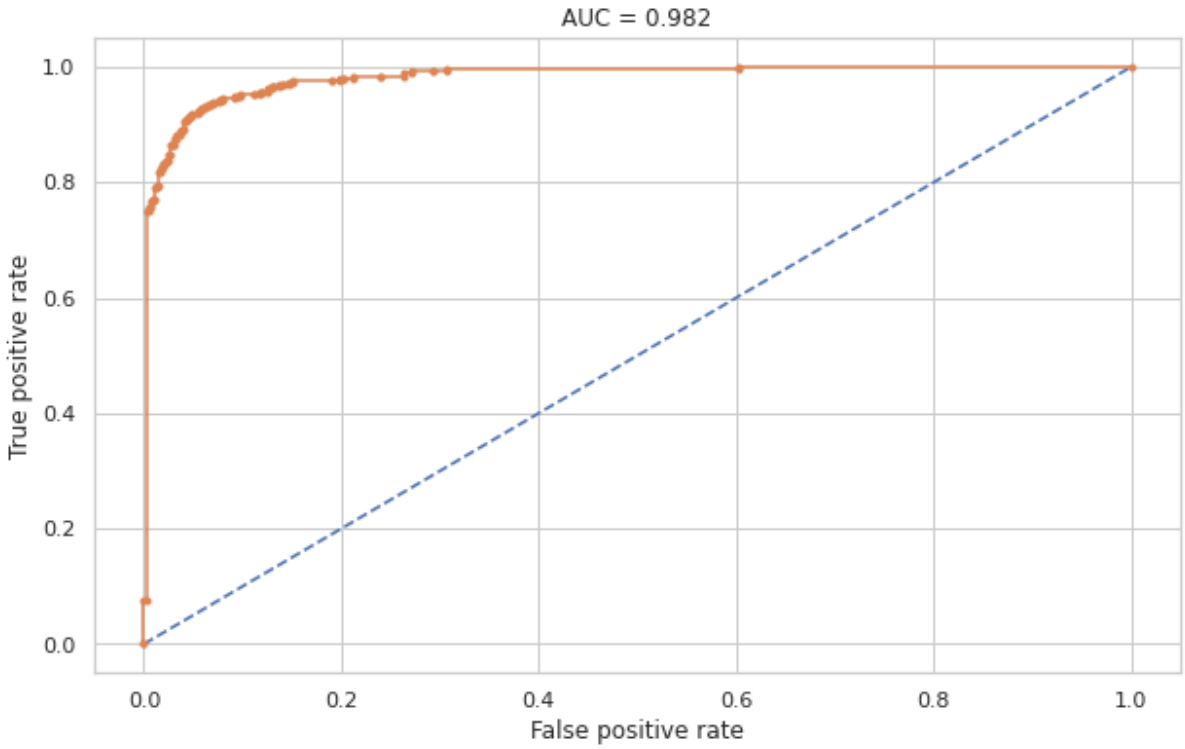
Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimised. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

Best parameters:
{'learning_rate': 0.1,
'max_depth': 11,
'n_estimators': 200}

Classification Report For This model is as follows

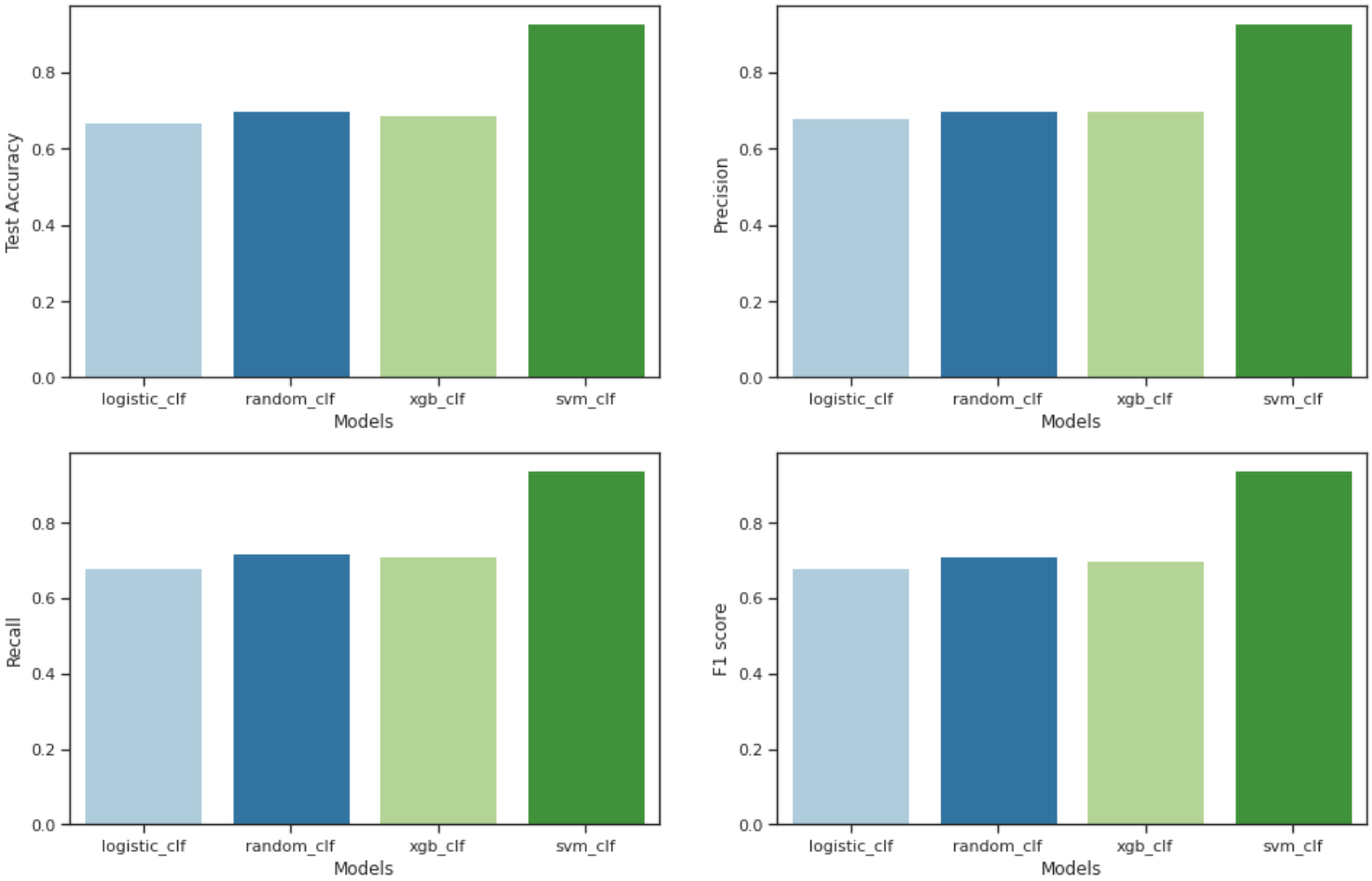
	precision	recall	f1-score	support
0	0.93	0.93	0.93	477
1	0.93	0.94	0.94	517
accuracy			0.93	994
macro avg	0.93	0.93	0.93	994
weighted avg	0.93	0.93	0.93	994

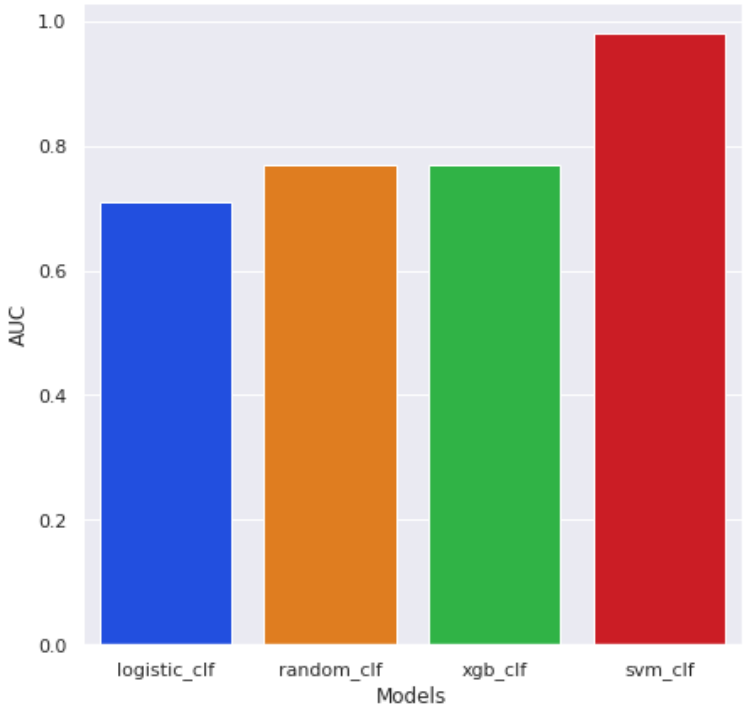




12.5 Evaluating Model Performances

This plot shows a comparison of all scores for applied models.





From above plots:

XGBoost, **Support vector machine** gives highest Accuracy, Recall, Precision and AUC score.

- Highest recall is given by Support vector machine
- Highest AUC is given by Support vector machine
- Overall we can say that the Support vector machine is the best model that can be used for the risk prediction of Cardiovascular heart disease.

From both the graphs we can say that the **best performing model** is **Support Vector Machine algorithm**.

Time taken by each algorithm is as follows:

• Logistic Regression	53.2 sec
• Random Forest	3 min 4 sec
• XGBoost	2 min 54 sec
• Support Vector Machine	14 min 35 sec

12. Conclusions

- **Risk** of **Cardiovascular heart disease** is almost **equal** between **smokers and non-smokers** and the same goes with **gender. It is** pretty much the same for both ***male*** and ***females***.
- **Correlation** obtained is **very poor** for this dataset but still due to **tuned parameters** and **strong** classification **algorithms** model **efficiency** obtained is about **93%**.
- The top **contributing features** in predicting the **ten year risk** of developing Cardiovascular Heart Disease are **'age', 'totChol', 'sysBP', 'diaBP', 'BMI', 'heartRate', 'glucose'**.
- The **Support vector machine** with the **radial kernel** is the **best performing** model in terms of **accuracy** and the **F1 score** and Its **high AUC-score** shows that it has a **high true positive rate**.
- **Balancing** the dataset by using the **SMOTE technique** helped in **improving** the **models' sensitivity**.
- With **more data** & with more correlated features **better models** can be built.