Heart Attack Risk Prediction Using Machine Learning

A

Project Report

Submitted for the partial fulfillment of B.Tech Degree in

COMPUTER SCIENCE & ENGINEERING

by

MANISH KUMAR PATEL

Department of Computer Science and Engineering

2024

Contents

<u>Declaration</u>

We hereby declare that this submission is our work and that, to the best of our belief and knowledge, it contains no material previously published or written by another person or material which to a substantial error has been accepted for the award of any degree or diploma of university or another institute of higher learning, except where the acknowledgement has been made in the text. The project has not been submitted by us at any other institute for the requirement of any other degree.

Submitted by: -   Date: 15 may, 2024

Name: **Manish Kuamr Patel**

Roll No.: SBU200041

Branch: Computer Science and Engineering

Abstract

Our project **"Heart Attack Risk Prediction"** is a system that uses the predictive capability of machine learning models based on similar data points collected in the past. It predicts the risk of a heart attack based on the data provided by the user regarding their physicality, symptoms, and medical history. Our project will estimate the risk of having a heart attack in the coming future. Our system takes input data and runs a Machine Learning Model on the data and gives the result. The aim is to provide users with remote access to screening facilities that are capable of identifying low and high-risk individuals, thereby reducing the load on the medical system.

Heart attack risk predictor is an online platform designed and developed to explore the path of machine learning. The goal is to predict the risk of a heart attack in a patient from collective data, to be able to detect configurations at risk for the patient, and therefore, in cases requiring emergency medical assistance, alert the appropriate medical staff of the situation of the latter.

We can anticipate the danger of a heart attack in our project by evaluating the data. Physicians can also benefit from machine learning algorithms that provide important stats, real-time data, and data analysis related to the patient's medical condition.

List of Figures

List of Tables

Chapter 1

Introduction

Heart attack is one of the major causes of sickness and death worldwide, with more mortality rate than any other disease each year. According to the WHO, around 18 mils. humans died from heart disease in 2016, accounting for 30% of all demises worldwide. Developing and underdeveloped countries accounted for the major part of these deaths.

When the flow of blood to the heart is substantially diminished or stopped, it results in a heart attack. When fat and cholesterol pile up in the heart (coronary) arteries, it causes blockage. Plaques are fatty, cholesterol-containing deposits. The formation of plaque is known as atherosclerosis.

Coronary heart diseases (also called heart attacks) are very common and fatal for all heart diseases. Every 40 seconds, a person dies of a heart attack in the United States of America. And almost 800,000 people suffer from heart attacks each year.

There is a lot of research work going on on the topic of Heart disease in the field of medical research which is a major cause of mortality among other life-threatening disorders. Heart disease detection is a sophisticated process that can provide an automated prediction of a heart's health in a patient so that they can seek better treatment. For diagnosing heart disease, the indications, symptoms, and medical checkups are commonly used. Some of the factors that greatly enhance the risk of a heart attack in a person are his cholesterol habits, smoking habits, lack of any physical activities, obesity, increased blood pressure, and any history of heart diseases in their family.

Heart diseases consist of several diseases under its category, such as heart rhythm issues(also known as arrhythmias), congenital heart defects, and blood vessel illnesses, like coronary artery disease.

The positive side is that some modest changes in the way of living(such as daily exercising, quitting smoking, and having a healthy diet) can easily avoid the risk of having a heart attack. These changes, when combined with early treatment, can greatly decrease the risk. Due to the multivariate nature of multiple factors that contribute to risk such as high blood pressure, diabetes, high cholesterol, and so on, it becomes a challenging task to identify patients with high risk.

Nowadays, scientists and doctors are using various machine learning algorithms to build screening tools because of their superiority to identify patterns and classification in contrast to traditional stats-based approaches.

Cardiovascular diseases are still a major cause of morbidity among people all over the world. Risk prediction is a tough task for medical practitioners as it is a difficult task that demands a

lot of information and experience in the medical field. An automated medical diagnosis system would improve medical efficiency while simultaneously lowering expenses. We'll create a system that can quickly identify the criteria for predicting a patient's risk level based on health-related factors. The goal is to employ techniques involving data mining in order to find important unseeable patterns in heart problems and to predict the risk of a heart attack in individuals. A large amount of data is required to predict heart diseases that are too complex to handle and evaluate while using standard methods.

Our goal is to create a good machine learning approach for predicting the risk of cardiac disease(heart attack) that is both computationally efficient and accurate.

Chapter 2

Literature Review

Data mining techniques were used by I Ketut Agung Enriko, Muhammad Suryanegara, and Dadang Gunawan [1] in predicting heart disease by simplifying the parameters that need to be used for use in M2M remote patient monitoring purposes. To boost accuracy, KNN was used with the parameter weighting approach. Because the parameters were quick and easy to measure so that they could be tested in the comfort of their home, just 8 of the necessary 13 parameters are employed. The results demonstrate that the accuracy of these 8 parameters using the KNN method is adequate, as compared to 13 parameters using the KNN, Naive Bayes, and Decision Tree.

A prediction system to identify cardiac disease using a patient's medical data collection should be developed, according to Ashwini Shetty A and Chandra Naik [2] 2016. 13 risk factors were used as inputs for the system. Data pre-processing and then, data integration were done after the initial study of data from the dataset. They used Neural Networks and Genetic algorithms to predict heart diseases. A system was created which used a historical cardiac database to give the patient's diagnosis. All of the 13 different parameters were used to make the system. To extract the knowledge from a database, data mining techniques can be utilized. The Cleveland Heart Database was used as a source of data. It had a total of 300 records with 13 characteristics. Based on these 13 factors, it will determine if the patient had heart disease or not.

Sultana, Haider, and Uddin [3] 2017 suggested a cardiovascular disease analysis. This study presented data mining strategies for heart disease prediction. It can prove to be an important asset for healthcare professionals as it will provide us with an outline of existing strategies for gaining valuable info from the dataset. The performance of the model could be measured by calculating the time used by the model to generate the decision tree. The motive behind this study was to predict the disease using a lesser number of characteristics.

Kirmani [4] 2017 proposed utilizing data mining approaches to forecast many diseases. Data mining is now broadly used to predict a variety of illnesses. The number of physical examinations of a patient can be lowered by utilizing various data mining techniques. This research focuses on Chaitali S. Danger's prediction of heart disease, diabetes 7, and breast cancer in 2012 using three distinct models: Naive Bayes, Decision Trees, and Neural Networks. He increased the number of features in the Cleveland database by two, bringing the total to 15. The data mining methods used for the classification such as Decision Trees and Neural Networks were applied to the Cleveland database.

Chaitrali and Sulabha [5] adopted a slightly different approach to predicting heart diseases. They added 2 more characteristics (obesity and smoking) to the widely used Cleveland Heart disease database which consisted of around 300 records. They used Decision Trees, Naive Bayes, and Neural Networks as the classification techniques for data mining.

A few investigations were recognized in the past and different machine learning models have been utilized for categorization and forecast of heart disease determination. A robotized classifier recognizes between those with tall and mild hazards of congestive heart disappointment. Melillo [6] used a machine learning approach called CART. It stands for Classification and Regression Trees and it achieved a sensitivity of around 93% sensitivity and a specificity of approximately 63%.

Guidi [7] then adds a clinical decision support system for detecting and preventing heart failure at an early stage. They compared support vector machines, random forests, and CART algorithms, as well as other machine learning and deep learning models, notably neural networks. With an accuracy of 87.6%, Random Forest and CART outperformed everyone else in classification.

Zhang [8] used a mix of natural language processing and a rule-based method to estimate the NYHA HF class from unstructured clinical notes with 93.37 percent accuracy.

The SVM techniques used by Parthiban and Srivatsa [9] for identifying diabetic patients and forecasting heart disease had a 94.60 percent accuracy rate, and the factors used were common such as blood sugar, patient age, sys BP, and dia BP.

Chapter 3

Methodology

3.1 Data Collection

The dataset is taken from a town called Framingham in the city of Massachusetts where cardiovascular research was being conducted. It is available to the public on the Kaggle website. The purpose of this project is to find out the risk associated with developing coronary heart disease(CHD) in the coming years time. if the patient has a risk of developing coronary heart disease (CHD) in the coming 10 years. The data collection contains information about the patients. There are almost 4000 records and 15 qualities in all. Each characteristic has the possibility of becoming a risk factor. Risk factors include demographic, behavioral, and medical concerns.

3.1.1 Attributes:

3.1.1.1. Demographic:

- Sex: gender of the patient(0 for female and 1 for male)

- Age: patient's age(Continuous — Even though the documented ages were rounded off to be integers, the age attribute is always taken as continuous)

3.1.1.2. Education: We have eliminated this column from the dataset during pre-processing because no further information is given.

3.1.1.3. Behavioral attributes:

- Current Smoker: the patient is having a smoking habit or not (Binary)

- Cigs Per Day: an average value of the number of cigarettes smoked every day by the individual (Can be deemed continuous as even half a cigarette, can be consumed)

3.1.1.4. Information on   medical history:

- BP Meds: if the patient was using blood pressure medication (Binary)

- Prevalent Stroke: if the patient has encountered a prior stroke (Binary)

- Prevalent Hyp: if the patient was hypertensive or not (Binary)

- Diabetes: The presence or absence of diabetes in the patient (Binary)

3.1.1.5.Current medical condition:

- Sys BP: systolic blood pressure of the patient

- Dia BP: diastolic blood pressure of the patient

- Tot Chol: cholesterol levels of a patient

- Heart Rate: heart rate (Discrete values such as heart rate are treated as continuous in medical research, discrete variables like heart rate are treated as continuous since there is a large number of potential values)

- Glucose: blood glucose level (Continuous)

- BMI: Body Mass Index (Continuous)

3.1.2 Target variable to predict:

The chance of acquiring the "Coronary Heart Disease" (CHD) in 10 years: ("1", denotes "There is a risk", "0" denotes "There is no risk")

framingham

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	BMI	heartRate	glucose	TenYearCHD
1																
2	1	39	4	0	0	0	0	0	0	195	106	70	26.97	80	77	0
3	0	46	2	0	0	0	0	0	0	250	121	81	28.73	95	76	0
4	1	48	1	1	20	0	0	0	0	245	127.5	80	25.34	75	70	0
5	0	61	3	1	30	0	0	1	0	225	150	95	28.58	65	103	1
6	0	46	3	1	23	0	0	0	0	285	130	84	23.1	85	85	0
7	0	43	2	0	0	0	0	1	0	228	180	110	30.3	77	99	0
8	0	63	1	0	0	0	0	0	0	205	138	71	33.11	60	85	1
9	0	45	2	1	20	0	0	0	0	313	100	71	21.68	79	78	0
10	1	52	1	0	0	0	0	1	0	260	141.5	89	26.36	76	79	0
11	1	43	1	1	30	0	0	1	0	225	162	107	23.61	93	88	0
12	0	50	1	0	0	0	0	0	0	254	133	76	22.91	75	76	0
13	0	43	2	0	0	0	0	0	0	247	131	88	27.64	72	61	0
14	1	46	1	1	15	0	0	1	0	294	142	94	26.31	98	64	0

Table 3.1 Framingham dataset

3.2 Data Cleaning and Pre-Processing

We examined the data set for missing and duplicate variables, as they might have a significant impact on the effectiveness of machine learning methods (many algorithms do not tolerate missing data). When we were going through the data pre-processing phase, we found that the

dataset had no duplicate rows in the dataset. Although some of them were missing some entries, which can be denoted via the following graph:
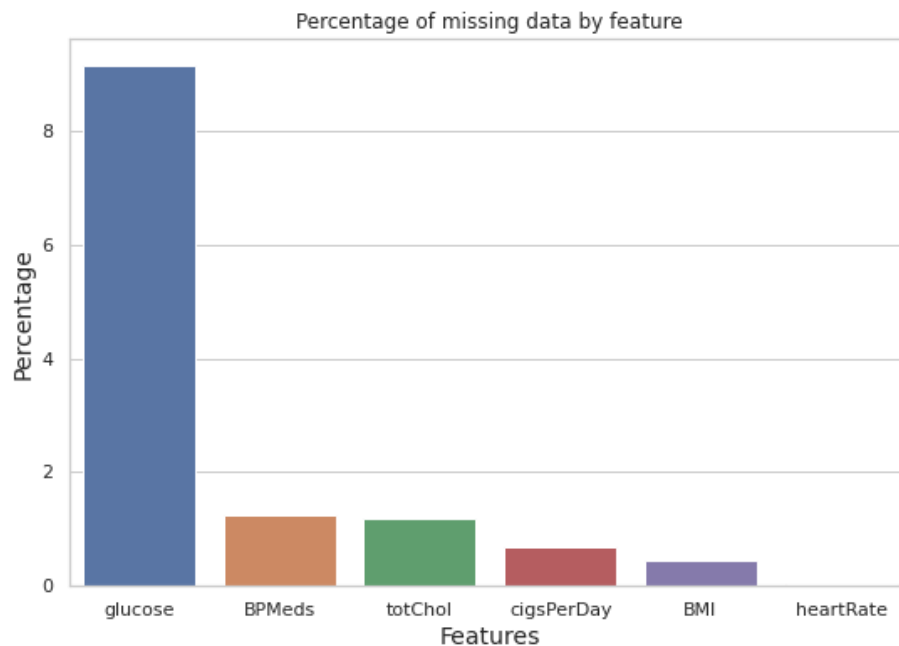

Fig-3.1:Missing data percentage by features

The blood glucose item had the largest amount of missing data, at 9.15 percent. There are relatively few missing entries in the other features. Because the rows that were missing some of the values comprised only 12.73% of the overall dataset, they were removed and there was not any significant data loss.

3.3 Dataset Analysis

We wanted to acquire crucial statistical insights from the data, so we looked at the distributions of the various qualities, their relationships with one another, and the objective variable, as well as calculating important odds and proportions for the categorical attributes. The initial stage was to examine the distribution of various qualities, which was best represented using histograms.
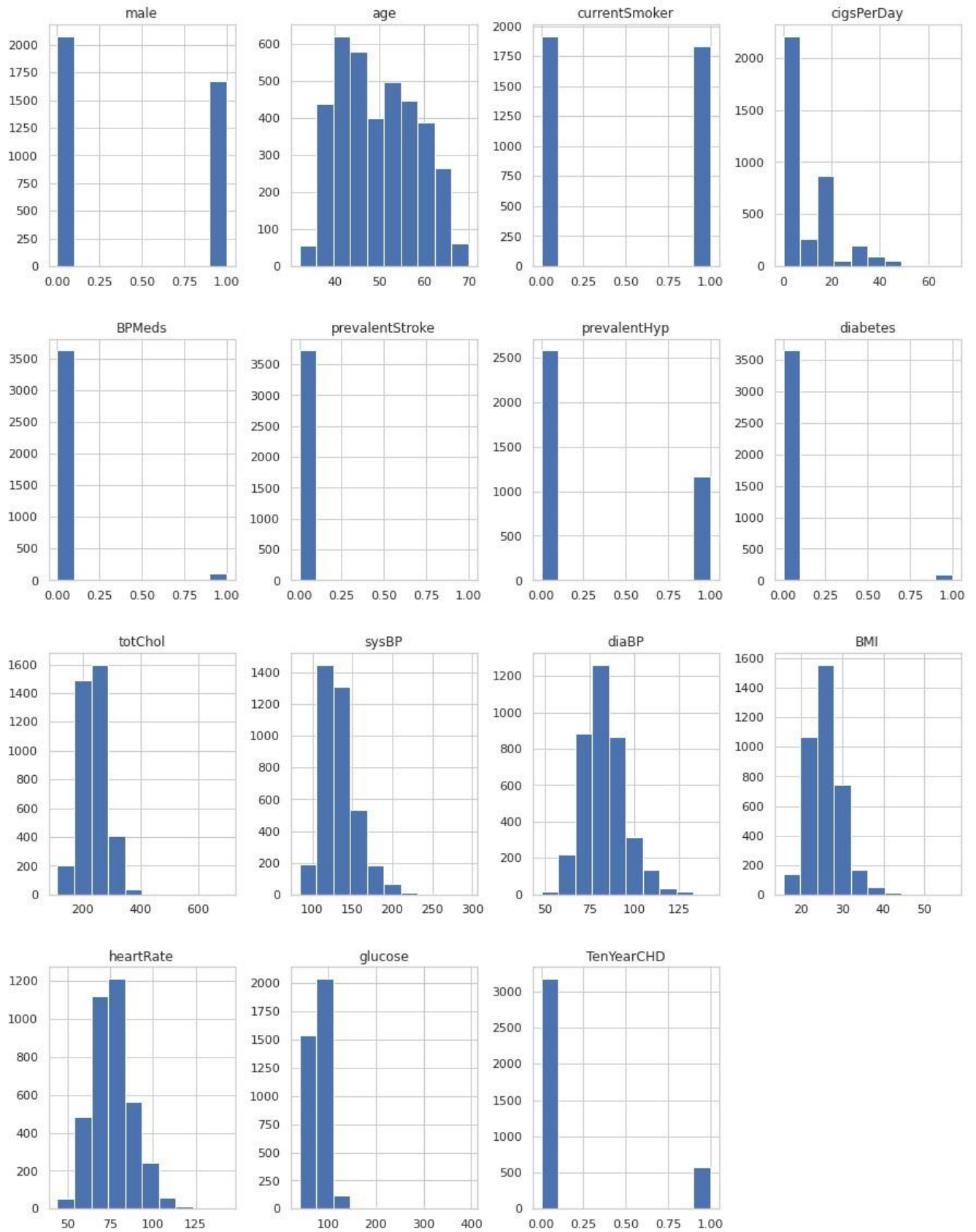
Fig-3.2:Dataset analysis histogram The distribution graphs identify categorical and continuous variables. In addition, none of

the participants had ever had a stroke, and just a rare number of patients were suffering from diabetes or hypertension, or they were on any blood pressure medication. The following plot also suggested that the dataset could be uneven, so we looked at the amount of positive and negative examples to confirm our doubts. The people that were suffering from CHD were 3179 and those with CHD were 572.
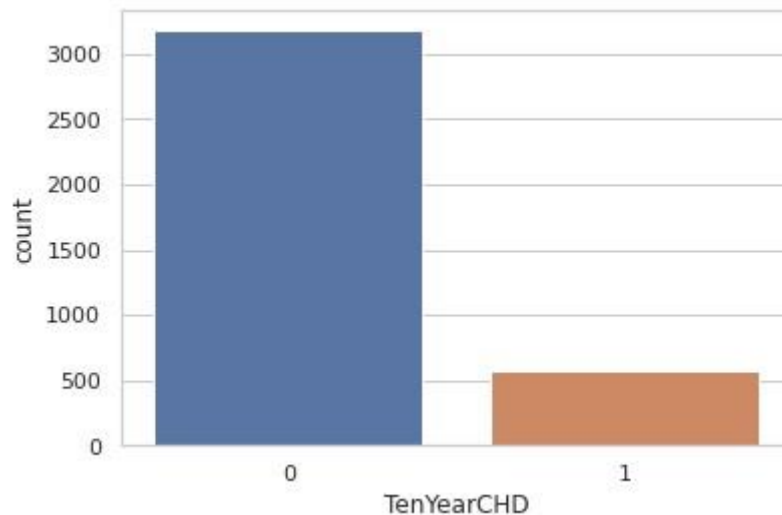
A fig-3.3:Initial positive and negative count

Since the dataset was unequal, it became challenging in order to conclude; yet, the following inferences may be implied based on what was observed from the given dataset:

- CHD affects men somewhat more than women.
- The percentage of patients having Coronary Heart Disease is similar between people who smoke and people who don't.
- Patients who have diabetes and prevalent hypertension, and are also having CHD have a higher percentage than the people who do not have similar medical conditions. ● People with CHD are more likely to use blood pressure medicine.

Another interesting trend we discovered was the spread of age among people who were suffering from CHD revealed that the number of patients with CHD increased with age, peaking at 63 years old.
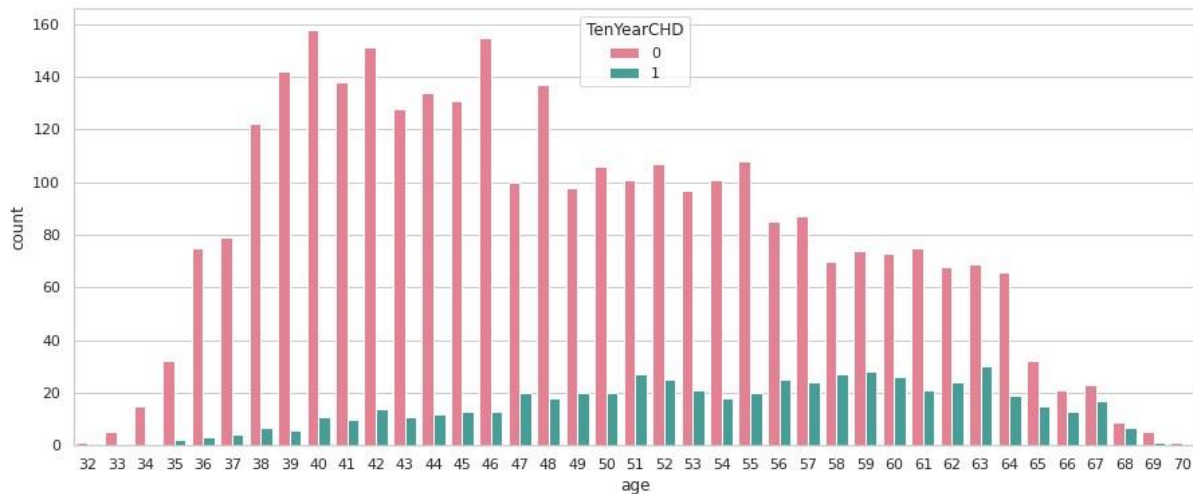
Fig-3.4: Age versus positive cases

The third stage was to examine various characteristics with the objective attribute and among themselves for any correlation since this would show the collinearity between the given features and also provide a fair assessment of the strength of attributes as the predictors of coronary heart disease.

Fig-3.5 Correlation heatmap

None of the characteristics in the matrix had a correlation > 0.5 with a ten-year chance of acquiring CHD implying that the characteristics were poor predictors of CHD. But, we found out that the correlation between prevalent hypertension, sys BP, and age was maximum.

Furthermore, there were a few more characteristics that were greatly associated with each other, so to use these similar attributes to make a machine learning model did not make any point. Sys BP and Dia BP; diabetes and glucose; a few more attributes are the numbers of cigarettes he/she smoked each day and the cigarette smoking habits of that person.

3.4 Handling Imbalanced Data

Imbalanced data distribution is a phrase used frequently in Machine Learning and Data Science to describe when observations in one class are considerably higher or lower than observations in other classes. Machine Learning algorithms ignore class distribution since their goal is to improve accuracy by decreasing error. Fraud detection, anomaly detection, and facial recognition are all examples of this challenge.

Decision Tree and Logistic Regression, for example, are common machine learning algorithms that favor the majority and ignore the minority. They tend to predict just the majority class, which leads to the false predictions of the minority class. Technically speaking, if our dataset has an uneven data distribution, our model is more susceptible to the circumstance where the minority class has a negligible or extremely low recall.

Unbalanced Data Handling Techniques:

There are two main techniques for dealing with an unequal distribution of classes.

1. SMOTE
2. Near Miss Algorithm

3.4.1 Synthetic Minority Oversampling Technique (SMOTE) – Oversampling

Whenever oversampling techniques come into discussion, SMOTE algorithm is the most extensively utilized technique to get rid of imbalance in the dataset (synthetic minority oversampling technique). Its purpose is to create minority class samples at random to equal out class distribution.

SMOTE creates new minority instances by combining existing minority instances. The Synthetic Minority Oversampling Approach (SMOTE) method uses the KNN technique to discover K-nearest neighbors, combine them, and generate synthetic samples in the space. The approach is used to calculate the distance between feature vectors and their nearest neighbors.

The difference is then multiplied by a randomly generated value ranging from 0 to 1 and returned to the attribute. SMOTE is a ground-breaking algorithm from which many others have been created.
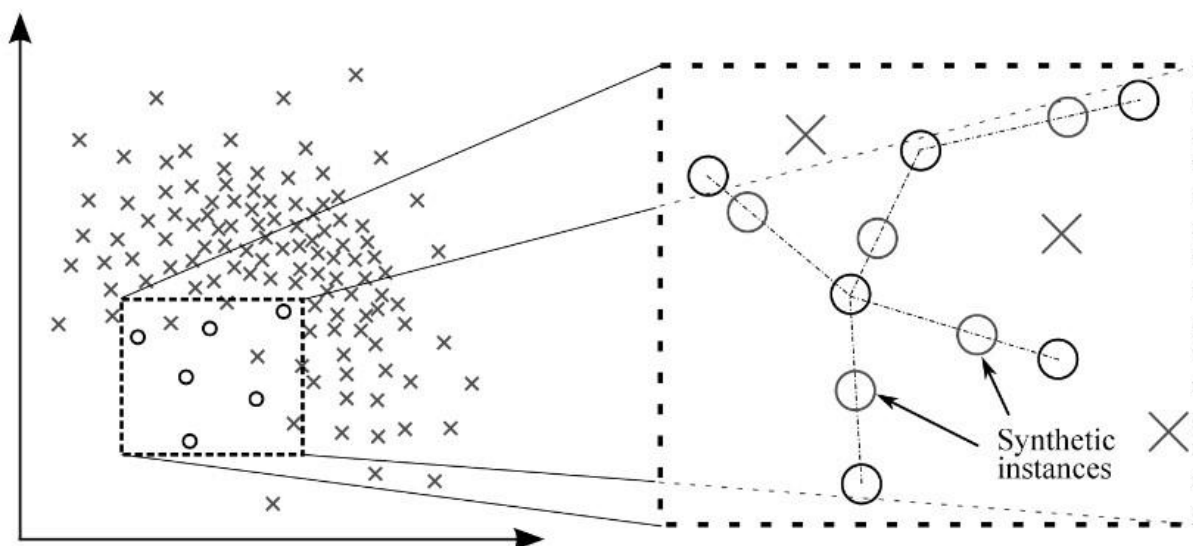

Fig-3.6 Smote algorithm

More details on the SMOTE Algorithm's operation!

Step 1: For each x ∈ A, to get the k-nearest neighbors of a point x, the Euclidean distance is calculated between point x and every other point in sample A.

Step 2: The uneven percentage is used to determine the sample rate N. N samples (i.e. x1, x2,...xn) are randomly picked from their k-nearest neighbors for each x ∈ A, and they form the collection A1.

Step 3: The following formula is used to produce a new example for each x_k ∈ A (k=1, 2,.....N):

$$x' = x + rand(0, 1) * \mid x - x_k \mid$$

The random integer between 0 and 1 is represented by rand(0, 1).

This method is used to generate randomly generated data points in minority class in the dataset. To balance the distribution, firstly the majority class is undersampled followed by the SMOTE algorithm to increase the instances in the dataset by oversampling the minority class.

The resulting data set was significantly more balanced after employing this approach, with 3178 negative cases and 2543 positive cases.
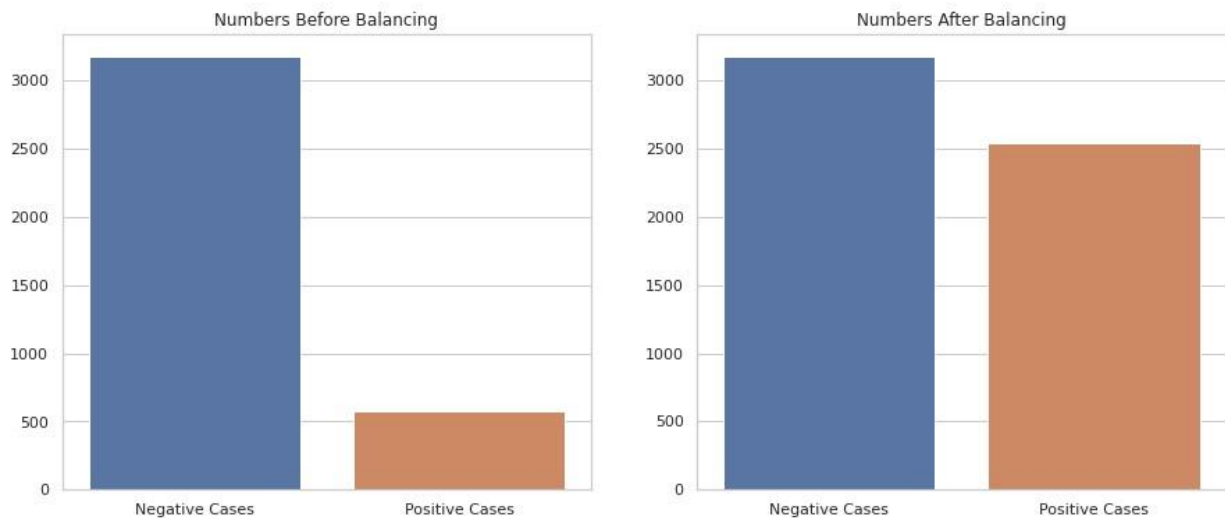
Fig-3.7 Count comparison after smote algorithm

3.5 Feature Selection

The results of the correlation matrix indicate that we need to use feature selection. To do the feature selection, the Boruta feature selection technique, a wrapper method that is built above the random forest classification, was used. Its main aim is to capture all the important and interesting elements of the dataset related to resultant variables.

3.5.1 Boruta Algorithm

Boruta algorithm can be listed in the following steps :

- An encrypted copy of all the features in the provided dataset is created, introducing unpredictability which is called the shadow feature.
- After the first step, it uses a larger dataset to train the random forest classifier, and then the boruta algorithm uses the feature importance metric (default of which is average reduction accuracy) to evaluate the value of each feature present in the dataset. The higher the score, the more important the characteristics are for our prediction.
- Then in each loop that is defined by the user, it analyzes whether the true feature is more important than its highest shadow feature (that is, in other words, whether the feature has a higher z-score than its corresponding shadow feature's maximum z-score) and is very irrelevant. Eliminate features that are considered irrelevant for the prediction.
- Boruta algorithm finishes after either all features are evaluated.

After that, we calculated the ratio of top features vs the chances of getting a heart attack, and we found the following data:

	5%	95%	Ratio
totchol	1.011381	1.033813	1.022536
glucose	0.994963	0.999184	0.997071
age	1.018236	1.031493	1.024843
heartRate	0.962258	0.984627	0.973378
BMI	0.929304	0.973798	0.951291
sysBP	0.963690	0.977730	0.970685
diaBP	1.001074	1.007518	1.004291

Table-3.2: Ratio of top features and positive count

3.5.1.1 Why is Variable Selection important?

- When we remove all the unnecessary variables then it will tend to improve the precision. The addition of any relevant variable has a quite similar effect on the model accuracy.
- Also, overfitting occurs when any model has too many variables and which as result is unable to generalize the pattern that forces it to overfit.
- Having a large number of variables which will eventually cause slow calculation, which in turn will require additional memory and hardware.

3.5.1.2 Why Boruta Package?

- All the classification and regression problems can be solved.
- Multivariable relationships which are correlated are considered in this algorithm.
- It's a better version of the random forest variable importance measure, which is one of the most popular variable selection methods.
- This algorithm can also deal with unpredictable interactions.
- Boruta algorithm can handle the most famous changing nature of the random forest significance measures.

3.5.1.3 Important points related to Boruta

- Before running the Boruta algorithm on our dataset, we made sure that any missing or blank values were filled in.
- After getting important variables from the Boruta algorithm, it's very tough to deal with collinearity.
- Slow speed is one of the main concerns as compared to other traditional feature selection techniques, it is slow.

3.5.1.4 Basic Idea of Boruta Algorithm

- If we talk about the basic idea of this algorithm then we can say that what it does is that it mixes the dataset with randomly permuted data of the original dataset and feeds it to random forest classifiers.

3.5.1.5 How Boruta Algorithm Works

To understand the algorithm, we can list it out it in a few steps that it follows -

- We create duplicate variables of the existing variables in the dataset.
- After that, we shuffle the dataset so that we can get rid of any correlation between the points.
- We then combine this data with the original dataset.
- After that, we run a random forest on a dataset and calculate the mean accuracy loss. We compare it with the created data.
- We calculate the Z score which is calculated by dividing the standard deviation by the mean of accuracy loss.
- Then we find the shadow variable with the highest Z score.
- If we get any variable Z score lower than the Z score calculated in the previous step then we can tag it as unimportant for that step and can look out for other variables.
- On the other hand, if it is higher then it is marked as important.
- We repeat the above steps until we get the priority of each feature.

3.5.1.6 Difference between Boruta and Random Forest Importance Measure

- We use the Z score to decide the feature importance in the boruta algorithm whereas we do not use it in random forest. Z score is calculated by dividing the standard deviation by the mean of the accuracy loss. And this value is related to the shadow feature so it makes boruta more reliable for the prediction of feature importance since it uses the random forest many times to get the actual importance, not a probabilistic one.

3.6 Machine Learning Classifier

KNN is majorly used in several types of research and development especially when there is very much less information about the data set is present or available also it is a very basic and pretty much state forward classification methodology. K nearest neighbor algorithm is said to be the non-parametric algorithm and by non-parametric, it means that there is not any kind of presupposition regarding the analyzed data's distribution that is followed in the model. This is useful in some situations where we get real-world data and it does not always follow the conceptual statistics, such as a normal distribution. K nearest neighbor algorithm is said to be a lazy algorithm it has less training period time also it does not create generalizations which simply means it keeps all of the training data that is fed.
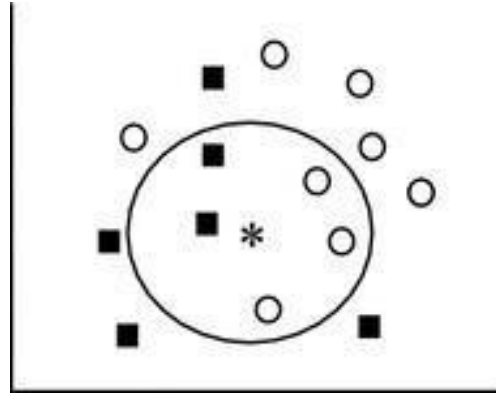
Fig-3.8: KNN (K nearest neighbour)

We can also say that it is a non-parametric classification technique. Its main function is to find the same dataset near to the current one and then classify based on the majority class. We have many ways to calculate the distance between the points and generally we use Euclidean distance to calculate the distance between two points. In the following figure, we can see the points that are distributed based on the distance and then classify one having more numbers is assigned as the final class of the particular points.
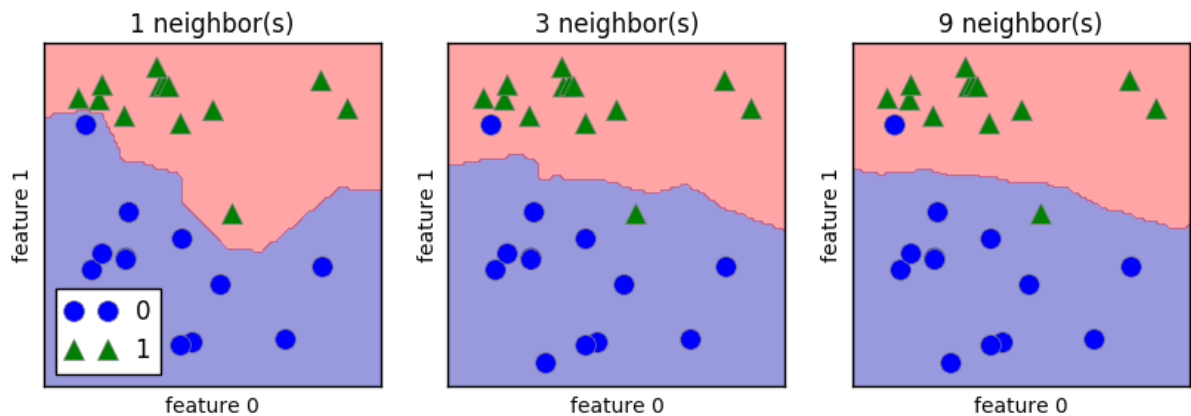

Fig-3.9: Difference in number of neighbors

Since, as we know that KNN is a lazy algorithm so it does not need much time while training, our main focus is to test out the model correctly since it is more important for the model to improve the accuracy with which it can predict the class of the incoming new data point based on the closest class.

We can also use KNN for regression problems. KNN works on the theory of nearest neighbours. So if anything is positive, it is very likely that points near it are positive. So in this way, we can use KNN for the regression problems and define the k such that it can select the closest points near to the given point. We will see the nearest neighbour concept in detail in the upcoming heading.

3.6.1 Nearest Neighbour:

We will try to understand the concept of nearest neighbour with the classification of two points in the two classes, that is whether a point is positive or negative.

We can see from the picture below that points that are close to a positive area are more likely to be positive in reality. So in KNN, we use this concept to decide in which class any particular point will go.

We calculate the nearest neighbour and assign the class of the most occurring class. If we see below then we can see that the point that is close to the negative side is most likely to be negative.
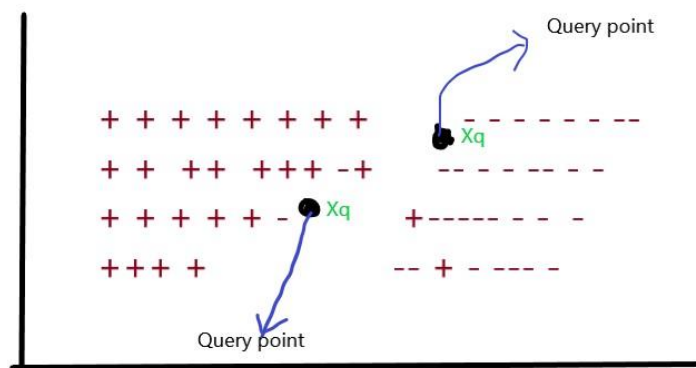

Fig-3.10: Classification using KNN

3.6.2 Distance Measures

We try to find the nearest neighbor of the query point using the distance between the given points and query points. When we talk in mathematics terms then we can say that distance calculation is the objective function that calculates the distance between two points in the same domain.

While talking theoretically we can say that it gives us the difference between both the points in the respective domain which is calculated using various methods that are present.

It is even possible to define this particular objective function by ourselves and can directly use the corresponding value for the satisfied condition. Such as we can say that if the first parameter is less than zero it must be counted as one else we can count it as zero. So it will work as our objective function which we can use to find the nearest neighbor using the KNN model. We

will see some of the most important and mostly used distance functions in the upcoming section.

3.6.2.1 Euclidean Distance

This is one of the most important and the most used distance functions. This is generally used to find the distance between two points where points are generally in int or float. We can see in the below picture that we have two points with the corresponding coordinates, we will find the distance between them, and that in turn will be used by the KNN model to find the K nearest neighbor.

One thing that we should focus on is that whatever data we are going to feed in while calculating the distance, it should be normalized to a normal scale to get the desired result.
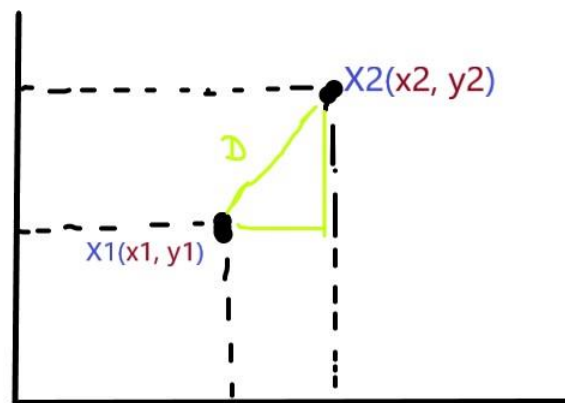

Fig-3.11: Euclidean distance

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

In the above picture, we can see the formula to find the distance between two points. We can also generalize the above one for any number of dimensions that we want to calculate. So basically in our model, we have a lot of features so it is always generalized to have the desired contribution while calculating the distance.

$$d = \sqrt{\sum_{i=1}^{N} (Xi - Yi)^{\wedge}2}$$

3.6.2.2 Manhattan Distance

This is one of the methods that we can use in our daily life also. It does not focus on the shortest path. Rather it takes the straight path possible to reach the points. We can see in the below picture that to calculate the distance between two points we have calculated the distance between the common points of intersection moving down and left. Then we add up these two distances to get the final distance between these two points. This method is used to find the K neighbor in the KNN model if we use this method as our distance calculation objective function.
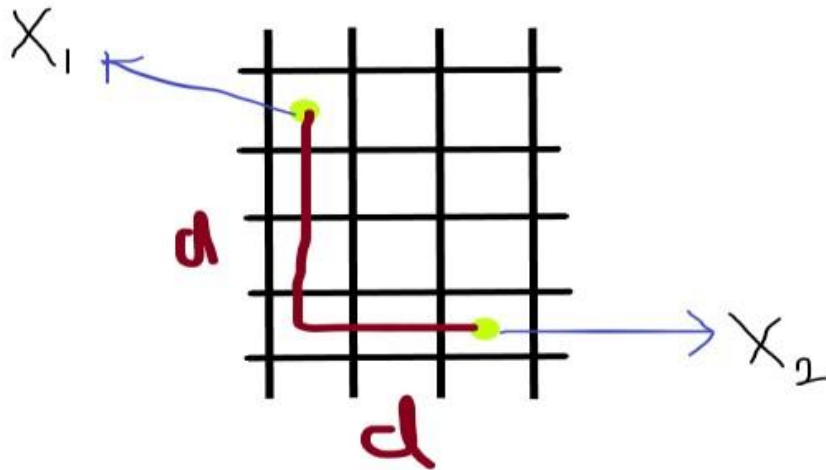

Fig-3.12: Manhattan distance

We can generalize it to the N dimension since we get multi-features in the machine learning models.

3.6.2.3 Minkowski distance

This method is the more generalized form of distance calculation and it takes care of both the approaches that Euclidean and Manhattan apply to calculate the distance between the points in the same dimensions.

When we say more generalized it means that we can use it for more features since the number of features in some models increases due to the importance of the features. We generally use this approach where we have got more features. This will also increase the calculation time complexity and would lead to more time in training and testing the dataset and also while predicting the actual values on the given query dataset.

We can write the formula mathematically for the Minkowski distance as below:

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^c \right)^{\frac{1}{c}}$$

3.6.2.4 Hamming distance

This method of measuring the distance is based on binary data. For more than one type of vector which is binary having only two digits, this method tries to count the number of vector points that are not the same.

This type of distance measuring method is different from the previous method in a way it deals with distinct binary strings having only two types of values and reports the number of positions where values are not the same.

Let's understand this with the help of the figure below. If we have two boolean strings with values of zero and one. According to this distance method, the answer is three.
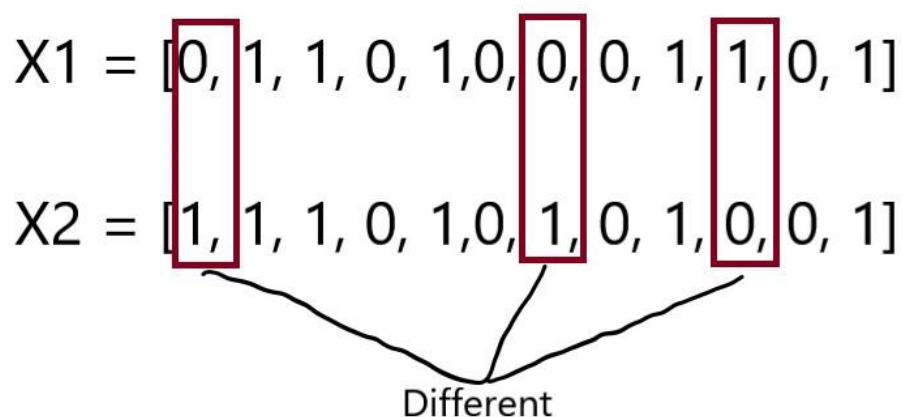
X1 = [0, 1, 1, 0, 1,0, 0, 0, 1, 1, 0, 1]

X2 = [1, 1, 1, 0, 1,0, 1, 0, 1, 0, 0, 1]

Different

Fig-3.13: Hamming distance

In the above example of binary vectors or binary strings, the total number of points where values are different from each other is three and we can probably state that the XOR of values of those points is 1 which again implies the difference in two binary values.

3.6.3 The KNN's steps are:

1 — Initially we will be getting data whose nature is unclassified.

2 — Now as a next step for finding the neighbor we will be calculating the distance between the new data point and the data point received through the first point. Depending upon the type of distance method the values could vary as well.

3 — After calculating the distance we will get K nearest point which will help us to choose the class of point in the upcoming steps;

4 — We choose the smallest K and then based on the count we decide the class;

5 — We give the class which have occurred most time in the resulted set;

6 —The class that we have selected in the previous step is given to the incoming point;

Below is the image where we can see the prediction.

```
[31] # age totChol sysBP diaBP BMI heartRate glucose
     h = [[39, 195, 106 ,70 , 26.97, 80, 77]]
     prediction = knn_clf.predict(h)
     print('You are safe. 😊 ') if prediction[0] == 0 else print('Sorry, You are on risk. 👽')

You are safe. 😊
```

```
[32] h = [[65, 150, 180, 70, 26.97, 80, 77]]
     prediction = knn_clf.predict(h)
     print('You are safe. 😊 ') if prediction[0] == 0 else print('Sorry, You are on risk. 👽')

Sorry, You are on risk. 👽
```

Fig-3.14: Prediction using classifier

Chapter 4

Experimental Results

We have used the confusion matrix for evaluating the accuracy, specificity, and sensitivity of our model. We have chosen this to get more idea about the prediction nature of our model. In other words, we can compare the sensitivity and specificity to have an idea of whether our model is predicting true positives more accurately or it is predicting the true negative more accurately. In the end, we can see the overall accuracy of our model.

Predicted 1 = positive,

Predicted 0 = negative,

TN = true negative.

TP = true positive,

FP = false positive,

FN = false negative,

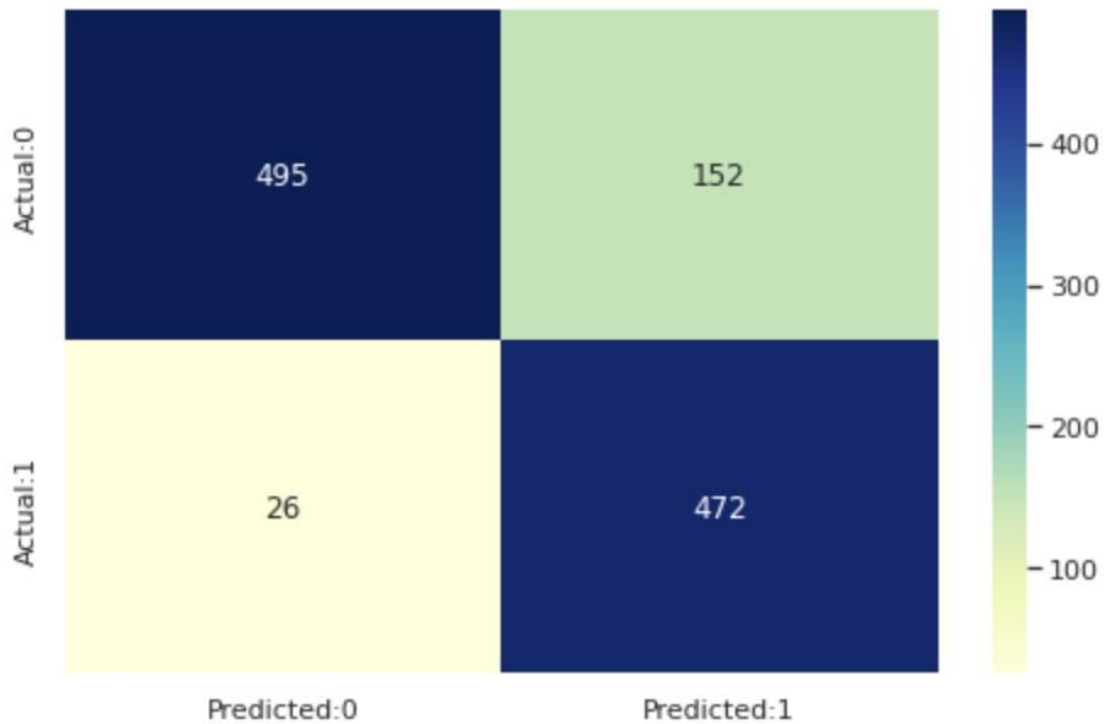|	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Fig-4.1: Confusion matrix

Fig-4.2: Result of confusion matrix

True Negative = 495

False Positive = 152

False Negative = 26

True Positive = 472

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

Accuracy achieved = 84.45%.

$$sensitivity = \frac{TP}{TP + FN}. \qquad specificity = \frac{TN}{TN + FP}.$$

Sensitivity = 94.77%.   Specificity = 76.51%.

```
# predictions
knn_predict = knn_clf.predict(X_test)
#accuracy
knn_accuracy = accuracy_score(y_test,knn_predict)
print(f"Using k-nearest neighbours we get an accuracy of {round(knn_accuracy*100,2)}%")
```

```
Using k-nearest neighbours we get an accuracy of 84.45%
```

Fig-4.3: Accuracy of our classifiers

Chapter 5

Conclusions

5.1 Conclusions

As we all are aware the number of deaths in today's world due to cardiovascular diseases and to be specific, heart attacks are gradually increasing. The issue needs to be found: how can we find a way or method so that we can predict diseases with a lot more accuracy and also with precision. The main aim of this project is that somehow we can predict the heart diseases in a person in 10 years. We have used different distance methods to find the closest neighbor and the K nearest neighbor algorithm which is also known as the lazy algorithm. For the feature selection method, we used Boruta which is an improvised version of the random forest method. For the dataset, we used UCI repository data. The result we get is around 84 percent accuracy with the help of the k-NN algorithm with hyperparameter tuning of the parameters. By tuning the parameters we can achieve higher accuracy.

Our models depict that age, sys BP, and BMI are the most important data here, followed by total cholesterol.
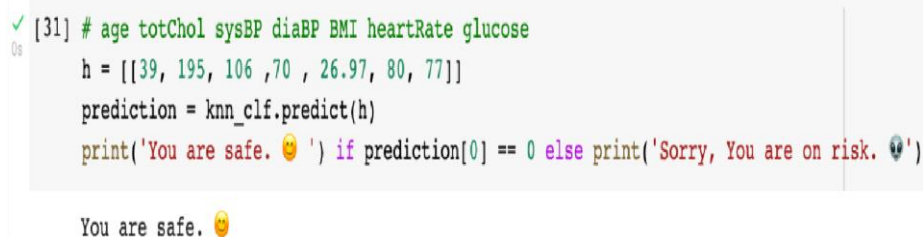
Let's use a random person as an example for the test.

A 39-year-old man with 195 total cholesterol, 106 systolic BP, 70 diastolic BP, a BMI of 26.97, a heart rate of 80, and glucose levels of 77.

Data sent to the backend for this particular person will be

[39, 195, 106, 70, 26.97, 80, 77]

After applying the machine learning classifier, we get the following result

```
[31] # age totChol sysBP diaBP BMI heartRate glucose
     h = [[39, 195, 106 ,70 , 26.97, 80, 77]]
     prediction = knn_clf.predict(h)
     print('You are safe. 😊 ') if prediction[0] == 0 else print('Sorry, You are on risk. 👽')

You are safe. 😊
```

Fig-5.1: Model prediction result

We can say with 84% accuracy that this person does not have a risk of having a heart attack in the next 10 years.

5.2 Future Works

In the future related to this project, we can develop a user interface that will help users to interact with our model and they can enter their details and can know the possibility of having a heart attack. Also, we can relate age and gender to the trend of having heart attacks in a person. One of the major problems that we faced during this project is the dataset. We can have more reliable sources of data that will lead to improved accuracy and that will eventually help us to predict more accurately.

After having a reliable source of data we can focus on the data handling part and normalize it to avoid the overfitting and underfitting of our model. In the future, more models can be evaluated and datasets can be fed and comparison can be done between the accuracy of different models. This will give higher chances of catching the heart attack in the earlier stage.

Also, we can include one more feature which will give suggestions based on age and other features to avoid a heart attack in the future. We can find any relation between the feature and the chances of having a heart attack in the future. We can then find out the stage of developing the symptoms that can lead to heart attack and then suggest different types of exercise and other activities that one should follow to avoid or decrease the risk of a heart attack in the future.

References

[1]   Enriko, I. K. A., Suryanegara, M., & Gunawan, D. (2016). Heart disease prediction system using k-Nearest neighbor algorithm with simplified patient's health parameters. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 8(12), 59-65.

[2]   Shetty, A., & Naik, C. (2016). Different data mining approaches for predicting heart disease. Int J Innov Res Sci Eng Technol, 5(9), 277-281.

[3]   Sultana, M., Haider, A., & Uddin, M. S. (2016, September). Analysis of data mining techniques for heart disease prediction. In 2016 3rd international conference on electrical engineering and information communication technology (ICEEICT) (pp. 1-5). IEEE.

[4]   Kirmani, M. M. (2017). Cardiovascular disease prediction using data mining techniques: A review. Oriental Journal of Computer Science & Technology, 10(2), 520-528.

[5]   Dangare, C. S., & Apte, S. S. (2012). Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications, 47(10), 44-48.

[6]   Melillo, P., De Luca, N., Bracale, M., & Pecchia, L. (2013). Classification tree for risk assessment in patients suffering from congestive heart failure via long-term heart rate variability. IEEE journal of biomedical and health informatics, 17(3), 727-733.

[7]   Guidi, G., Pettenati, M. C., Melillo, P., & Iadanza, E. (2014). A machine learning system to improve heart failure patient assistance. IEEE journal of biomedical and health informatics, 18(6), 1750-1756.

[8]   Zhang, R., Ma, S., Shanahan, L., Munroe, J., Horn, S., & Speedie, S. (2017, November). Automatic methods to extract New York heart association classification from clinical notes. In 2017 ieee international conference on bioinformatics and biomedicine (bibm) (pp. 1296-1299). IEEE.

[9]   Parthiban, G., & Srivatsa, S. K. (2012). Applying machine learning methods in diagnosing heart disease for diabetic patients. International Journal of Applied Information Systems, 3(7), 25-30.