A Capstone Project Report on

CARDIO VASCULAR DISEASE PREDICTION USING KNN

Submitted by

KESAVA SAI RAAM C N

ABSTRACT

As the human population increases, so is the chance of getting diseases. There are many illnesses globally, and one of the biggest problems faced by the hospital systems today is the lack of technology to know when the patients are ill. One such illness is Cardiovascular Disease or CVD. It refers to any heart disease, vascular disease, or blood vessel disease. According to WHO, more people die of CVD's worldwide than any other cause. It affects the low and middle-income countries more. It is very hard for people living alone to contact the hospital when they are sick. Therefore, we have developed a model that can detect when a patient is ill and report back to the hospital. The system currently only identifies patients with heart disease and reports back to the hospital. We decided to go with heart disease identification because it is one of the most deadly diseases, and the risk of patients dying because of heart disease is high. Predicting whether a patient has heart disease or not is very clearly a classification problem. Therefore, we have used five models to classify. We take several factors such as blood sugar level, age, cholesterol level, and many more and give the outcome based on the input.

Dataset and Source Code link: https://github.com/kesavasairaam/Capstone-Project.git

ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of my capstone project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, I have fortunate to have Mr. PRASAD as my mentor. He has readily shared his immense knowledge in data science and guides me in a manner that the outcome resulted in enhancing my data skills.

I certify that the work done by me for conceptualizing and completing this project is original and authentic.

Date: July 10, 2022 Name: KESAVA SAI RAAM C N

CERTIFICATE OF COMPLETION

I hereby certify that the project titled “Cardio Vascular disease prediction using KNN”was undertaken and completed the project (10^{th} July, 2022).

Mentor : Mr. Prasad

Date : 10^{th} July,2022

Place : Karur

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Cardiovascular disease is the leading cause of death worldwide and a major public health concern. Therefore, its risk assessment is crucial to many existing treatment guidelines. Risk estimates are also being used to predict the magnitude of future cardiovascular disease mortality and morbidity at the population level and in specific subgroups to inform policymakers and health authorities about these risks. Additionally, risk prediction inspires individuals to change their lifestyle and behaviour and to adhere to medications. Although several risk prediction models of cardiovascular disease have been developed for different populations in the past decade, the validity of these models is a cause of concern. Most data for model formation and validation have been provided from a small set of populations, mostly from developed countries

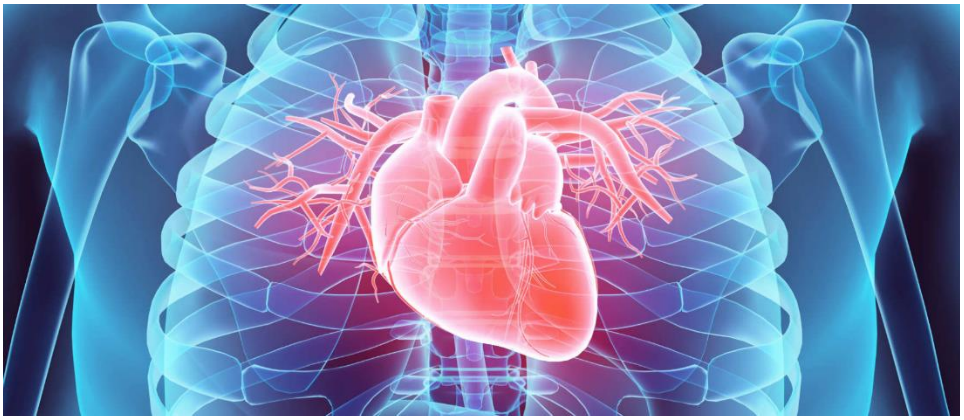

Figure 1 Cardio vascular disease

Therefore, using this set for the classification of individuals from different risk groups of other populations might lead to risk overestimation. This, in turn, can result in increased costs of guidelines and health interventions. These models might also cause risk underestimation, which can lead to missing vulnerable cases. Consequently, providing a valid model for cardiovascular disease risk classification of each population has become a high priority for scientists and organisations working in this field.

Heart Diseases have shown a tremendous hit in this modern age. As doctors deal with precious human life, it is very important for them to be right their results. Thus, an application was developed which can predict the vulnerability of heart disease, given basic symptoms like age, gender, pulse rate, resting blood pressure, cholesterol, fasting blood sugar, resting electrocardiographic results, exercise induced angina, ST depression ST segment the slope at peak exercise, number of major vessels colored by fluoroscopy and maximum heart rate achieved. This can be used by doctors to re heck and confirm on their patient's condition.

CHAPTER 2

DATA COLLECTION AND DATA PREPARATION

Dataset for cardiovascular disease is found from various sources such as from Kaggle, Medical API, web and browser searches. Finally, there are nearly 5395 details of various patients which is in csv format that contains all the basic datas to predict the disease of the patient such as age, gender, pulse rate, resting blood pressure, cholesterol etc.

Data description

There are 3 types of input features:

- Objective: factual information;
- Examination: results of medical examination;
- Subjective: information given by the patient.

Traditional cardiovascular risk factors often assessed in an annual physical, such as blood pressure, cholesterol levels, diabetes, and smoking status, are at least as valuable in predicting who will develop coronary heart disease (CHD) as a sophisticated genetic test that surveys millions of different points in DNA.

Features of the dataset:

1. Age | Objective Feature | age | int (days)
2. Height | Objective Feature | height | int (cm) |
3. Weight | Objective Feature | weight | float (kg) |
4. Gender | Objective Feature | gender | categorical code |
5. Systolic blood pressure | Examination Feature | ap_hi | int |
6. Diastolic blood pressure | Examination Feature | ap_lo | int |
7. Cholesterol | Examination Feature | cholesterol | 1: normal, 2: above normal, 3: well above normal |
8. Glucose | Examination Feature | gluc | 1: normal, 2: above normal, 3: well above normal |
9. Smoking | Subjective Feature | smoke | binary |
10. Alcohol intake | Subjective Feature | alco | binary |
11. Physical activity | Subjective Feature | active | binary |
12. Presence or absence of cardiovascular disease | Target Variable | cardio | binary |

All of the dataset values were collected at the moment of medical examination.

```
[ ] data.describe()
```

	age	female	male	height	weight	bmi	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active
count	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000
mean	53.302869	0.650429	0.349571	164.359229	74.205690	27.556545	128.817286	96.630414	1.366871	1.226457	0.088129	0.053771	0.803729
std	6.754974	0.476838	0.476838	8.210126	14.395757	6.091405	154.011419	188.472530	0.680250	0.572270	0.283484	0.225568	0.397179
min	29.560000	0.000000	0.000000	55.000000	10.000000	3.470000	-150.000000	-70.000000	1.000000	1.000000	0.000000	0.000000	0.000000
25%	48.360000	0.000000	0.000000	159.000000	65.000000	23.880000	120.000000	80.000000	1.000000	1.000000	0.000000	0.000000	1.000000
50%	53.940000	1.000000	0.000000	165.000000	72.000000	26.375000	120.000000	80.000000	1.000000	1.000000	0.000000	0.000000	1.000000
75%	58.390000	1.000000	1.000000	170.000000	82.000000	30.220000	140.000000	90.000000	2.000000	1.000000	0.000000	0.000000	1.000000
max	64.920000	1.000000	1.000000	250.000000	200.000000	298.670000	16020.000000	11000.000000	3.000000	3.000000	1.000000	1.000000	1.000000

Figure 2 Training Dataset

The mean age for patients is 53. The percentage of males is 35. The percentage of females is 65. The percentage of smokers is 8. The percentage of alcoholists is 5. The percentage of patients who do sports is 80. It seems there are many outliers in body mass index, maybe it's a mistake. So, lets drop outliers.

DROPPING THE OUTLIERS:

Created a function that adds a column called bp_cat (Blood Pressure Category). This function scans two columns of each row which are the ap_hi and ap_lo then based on the values of these columns it categorizes the patient’s blood pressure.

```
def BPCategorize(x,y):
    if x<=120 and y<=80:
        return 'normal'
    elif x<=129 and y<=80:
        return 'elevated'
    elif x<=139 or y<=89:
        return 'high 1'
    elif x<=180 or y<=120:
        return "high 2"
    elif x>180 or y>120:
        return 'high 3'
    else:
        return None

data.insert(8, "bp_cat", data.apply(lambda row: BPCategorize(row['ap_hi'], row['ap_lo']), axis=1))
data['bp_cat'].value_counts()
```

```
normal      39008
high 1      15380
high 2      15023
elevated      419
high 3         77
Name: bp_cat, dtype: int64
```

We can also drop outliers from blood pressure variables

```
[ ] data.drop(data.query('ap_hi >220 or ap_lo >180 or ap_hi<40 or ap_lo<40').index, axis=0, inplace=True)
```

Figure 3 Function for Blood pressure

CHAPTER 3

DATA PREPROCESSING

Data preprocessing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. When it comes to creating a Machine Learning model, data preprocessing is the first step marking the initiation of the process. Typically, real-world data is incomplete, inconsistent, inaccurate (contains errors or outliers), and often lacks specific attribute values/trends. This is where data preprocessing enters the scenario – it helps to clean, format, and organize the raw data, thereby making it ready-to-go for Machine Learning models.

```
data.head()
```

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0

Figure 4 Columns in the dataset

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   id           70000 non-null  int64  
 1   age          70000 non-null  int64  
 2   gender       70000 non-null  int64  
 3   height       70000 non-null  int64  
 4   weight       70000 non-null  float64
 5   ap_hi        70000 non-null  int64  
 6   ap_lo        70000 non-null  int64  
 7   cholesterol  70000 non-null  int64  
 8   gluc         70000 non-null  int64  
 9   smoke        70000 non-null  int64  
 10  alco         70000 non-null  int64  
 11  active       70000 non-null  int64  
 12  cardio       70000 non-null  int64  
dtypes: float64(1), int64(12)
memory usage: 6.9 MB
```

Figure 5 Information of the dataset

HANDLING THE MISSING VALUES

```
data.isnull().sum(axis = 0)
```

```
id             0
age            0
gender         0
height         0
weight         0
ap_hi          0
ap_lo          0
cholesterol    0
gluc           0
smoke          0
alco           0
active         0
cardio         0
dtype: int64
```

Figure 6 Missing the data values in the dataset

There is no missing values present in the dataset.

DATA CLEANING

The patient’s age is written in days, so we're converting it to years and rounding it to the nearest 2 decimals. Also we're replacing the gender column with another two-columns, one for male and the other is for female. If the patients' gender is male then a value of 1 will be inside the male column and zero inside the female column and vice-versa.

```
data.insert(3, "female", (data['gender']==1).astype(int))
data.insert(4, 'male', (data['gender']==2).astype(int))
data.drop(['gender', 'id'], axis=1, inplace=True)
```

Figure 7 Cleaning the data

We’re calculating the patient BMI (Body Mass Index) using the formula which is: weight/height^2.

In our dataset, the height of patients were in centimetres so we divided it by 100 to convert it into meters.

```
[ ]  data.insert(5, 'bmi', round((data['weight']/(data['height']/100)**2), 2))
```

EXPLORATORY DATA ANALYSIS

In statistics, exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

```
data.describe()
```

	age	female	male	height	weight	bmi	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active
count	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000
mean	53.302869	0.650429	0.349571	164.359229	74.205690	27.556545	128.817286	96.630414	1.366871	1.226457	0.088129	0.053771	0.803729
std	6.754974	0.476838	0.476838	8.210126	14.395757	6.091405	154.011419	188.472530	0.680250	0.572270	0.283484	0.225568	0.397179
min	29.560000	0.000000	0.000000	55.000000	10.000000	3.470000	-150.000000	-70.000000	1.000000	1.000000	0.000000	0.000000	0.000000
25%	48.360000	0.000000	0.000000	159.000000	65.000000	23.880000	120.000000	80.000000	1.000000	1.000000	0.000000	0.000000	1.000000
50%	53.940000	1.000000	0.000000	165.000000	72.000000	26.375000	120.000000	80.000000	1.000000	1.000000	0.000000	0.000000	1.000000
75%	58.390000	1.000000	1.000000	170.000000	82.000000	30.220000	140.000000	90.000000	2.000000	1.000000	0.000000	0.000000	1.000000
max	64.920000	1.000000	1.000000	250.000000	200.000000	298.670000	16020.000000	11000.000000	3.000000	3.000000	1.000000	1.000000	1.000000

Figure 8 Explore the data

It seems there are many outliers in body mass index, may be it's a mistake. So, lets drop outliers.

```
data.drop(data.query('bmi >60 or bmi <15').index, axis=0, inplace=True)
```

CHAPTER 4

DATA VISUALISATION

The dataset has been cleaned and sorted according to our needs by removing all the noisy outliers. Now visualize the data.

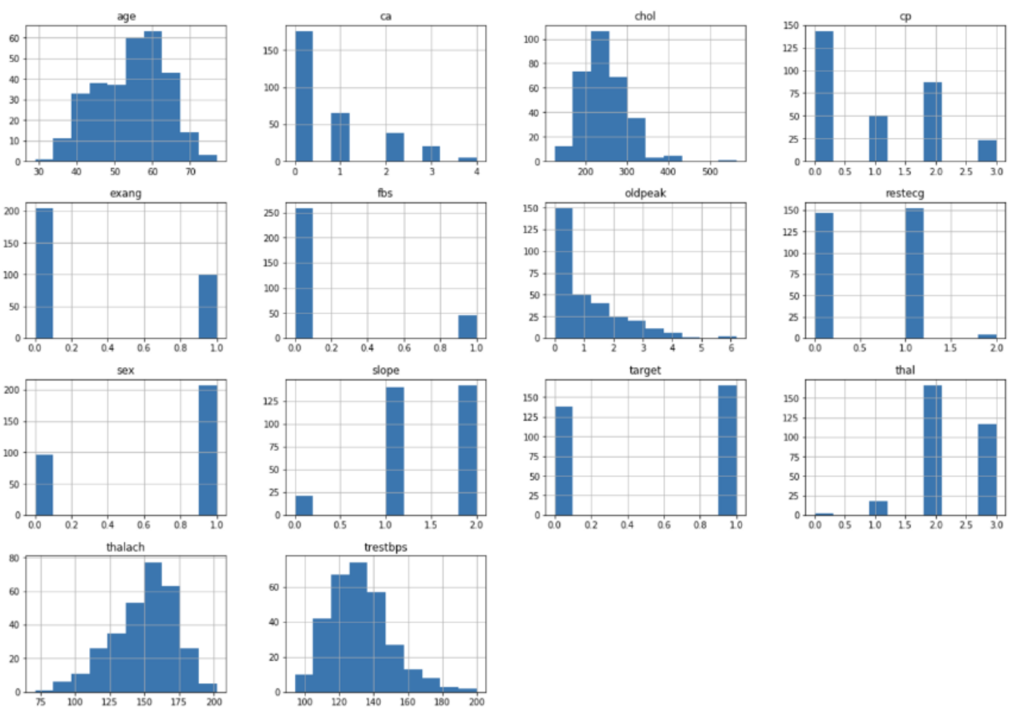

Figure 9 Histogram of the dataset

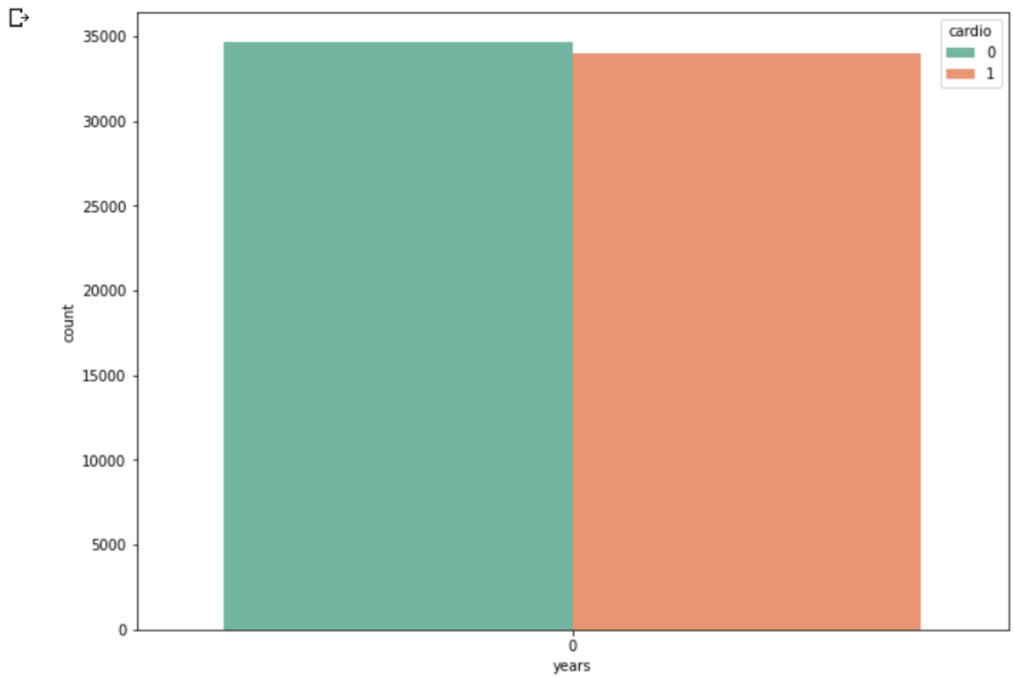

Figure 10 Visualizing of ages

CHAPTER 5

UNDERSTANDING THE DATA

Making boxplots to compare the age and body mass index for the cardio and non-cardio patients.

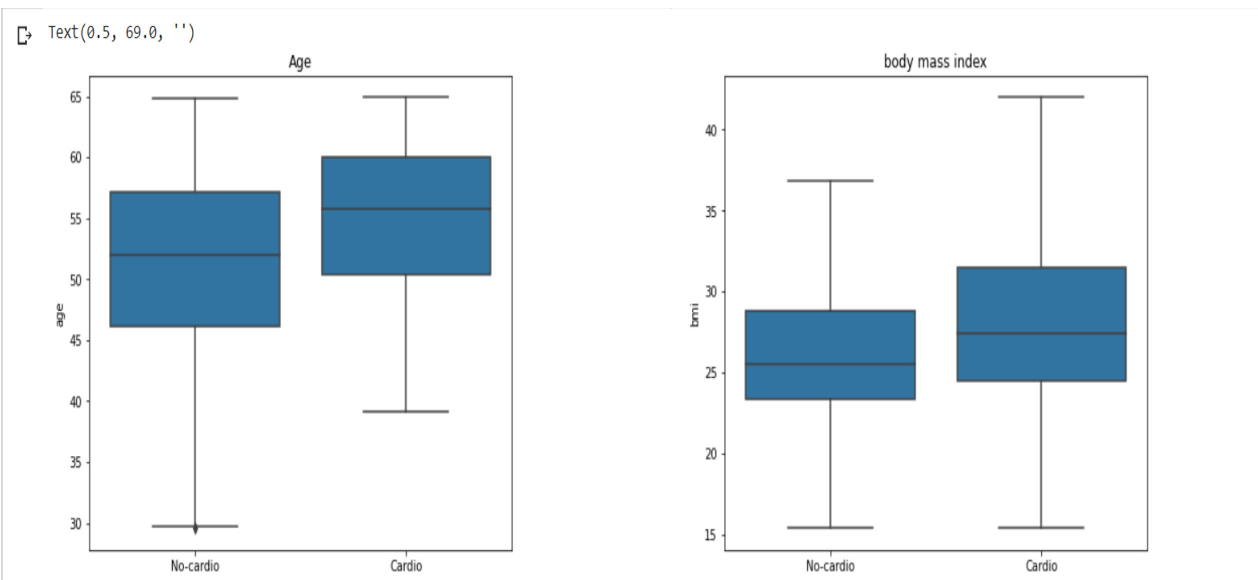

Figure 11 Boxplot

A relation is found between the age of people and cardiovascular diseases, thus, elderly people are most likely to have this kind of diseases. Another relation is found between the BMI and cardiovascular diseases, thus, people with higher BMI are also most likely to to have this kind of diseases.

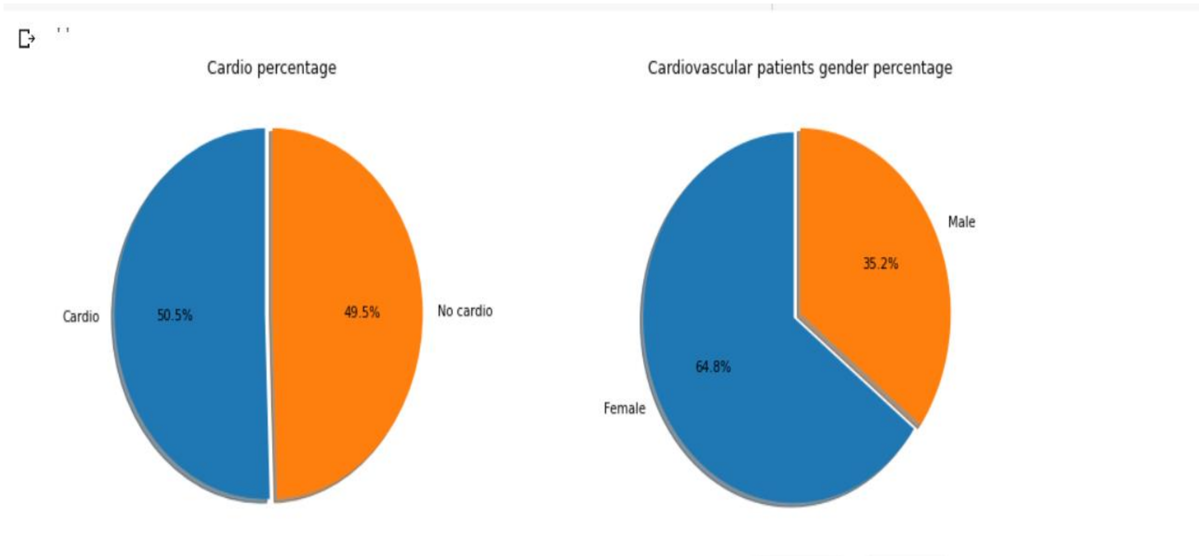

Figure 12 Piechart of gender

The percentage of people with cardiovascular diseases is 50%. The percentage of males with cardiovascular diseases is 35.3%. The percentage of females with cardiovascular diseases is 64.7%.

CHAPTER 6

PREDICTING NEW DATA

Make predictions:

Step 1: Data collection. The selection and preparation of data to train the system is one of the most important tasks in the process. ...

Step 2: Create a model ("train" the system) After the Dataset was created, we will create and train the model. ...

Step 3: Make predictions.

I have verified and get more values and percentages by using a single column by using the concept of probability.

Figure 6.1: Represents the probability that a person has cardio diseases given that he is 50 or older.

Figure 6.2: Represents the probability that a person drinks alcohol or smokes.

Figure 6.3: Represents the probability that a person has cardio diseases given that the patient drinks alcohol or smokes.

Predictive analytics involves certain manipulations on data from existing data sets with the goal of identifying some new trends and patterns. These trends and patterns are then used to predict future outcomes and trends. By performing predictive analysis, we can predict future trends and performance. It is also defined as the prognostic analysis, the word prognostic means prediction. Predictive analytics uses the data, statistical algorithms and machine learning techniques to identify the probability of future outcomes based on historical data.

```
[ ] data_age_50 = data.query('age >=50')
    data_agy_50_cardio = data_age_50.query('cardio==1')
    round(data_agy_50_cardio.shape[0]*100/data_age_50.shape[0],2)

    55.42
```

Figure 13 Probability of person

```
data_cohol_smoke = data.query("alco==1 or smoke==1")
print(data_cohol_smoke.shape[0]*100/data.shape[0])

11.524964689779694
```

Figure 14 Probability of person Smokes or drinks alcohol

```
data_cohol_smoke_cadrio = data_cohol_smoke.query('cardio==1')
data_cohol_smoke_cadrio.shape[0]*100/data_cohol_smoke.shape[0]
```

```
47.95957043588124
```

Figure 15 Probability of person has cardio disease who

Naive Bayes is a simple but surprisingly powerful algorithm for predictive modeling. The model consists of two types of probabilities that can be calculated directly from your training data: 1) The probability of each class; and 2) The conditional probability for each class given each x value

CHAPTER 7

TRAINING THE MODEL

K Neighbors Classifier

This classifier looks for the classes of K nearest neighbors of a given data point and based on the majority class, it assigns a class to this data point. However, the number of neighbors can be varied. I varied them from 1 to 20 neighbors and calculated the test score in each case.

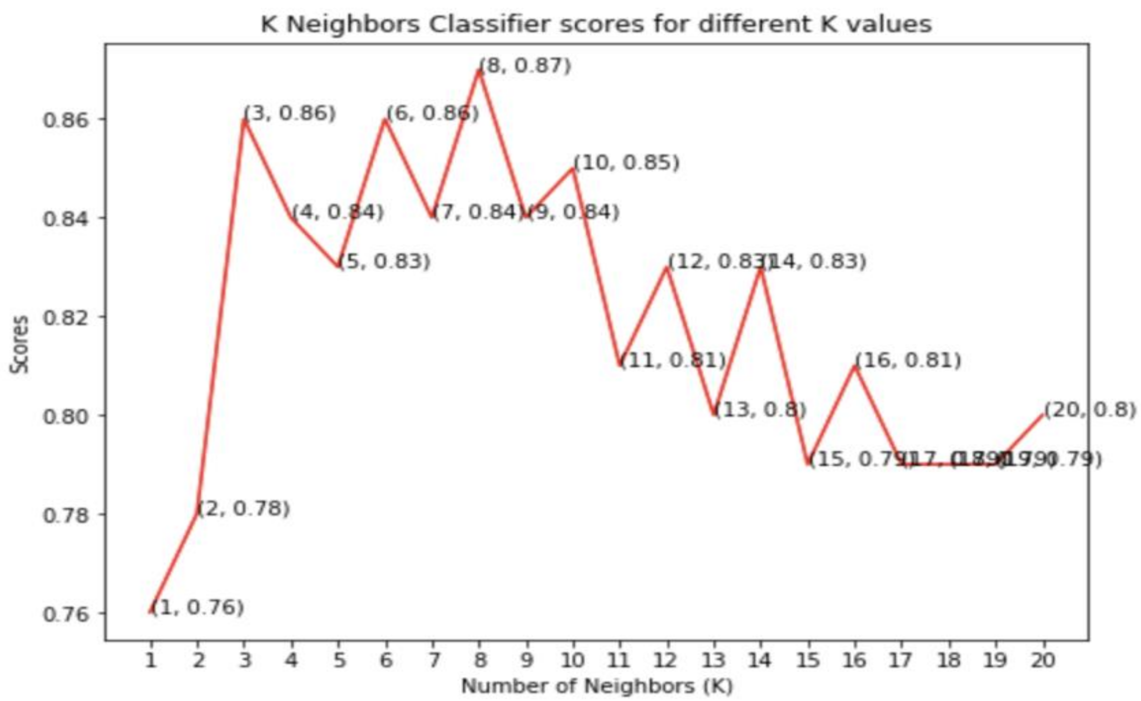

Figure 16 K neighbors' plot

The K in the name of this classifier represents the k nearest neighbors, where k is an integer value specified by the user. Hence as the name suggests, this classifier implements learning based on the k nearest neighbors.

Gaussian

Gaussian Processes are a generalization of the Gaussian probability distribution and can be used as the basis for sophisticated non-parametric machine learning algorithms for classification and regression.

Since Gaussian processes let us describe probability distributions over functions we can use Bayes' rule to update our distribution of functions by observing training data.

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.20, random_state = 0)

[ ] from sklearn.preprocessing import StandardScaler
    sc = StandardScaler()

    from sklearn.preprocessing import StandardScaler
    sc = StandardScaler()
    X_train = sc.fit_transform(X_train)
    X_test = sc.transform(X_test)

[ ] from sklearn.naive_bayes import GaussianNB
    classifier = GaussianNB()

[ ] classifier.fit(X_train, y_train)

    GaussianNB()
```

Figure 17 Gaussian Algorithm

Naive Bayes is a probabilistic machine learning algorithm used for many classification functions and is based on the Bayes theorem. Gaussian Naïve Bayes is the extension of naïve Bayes.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

<matplotlib.axes._subplots.AxesSubplot at 0x7fb98ce35310>

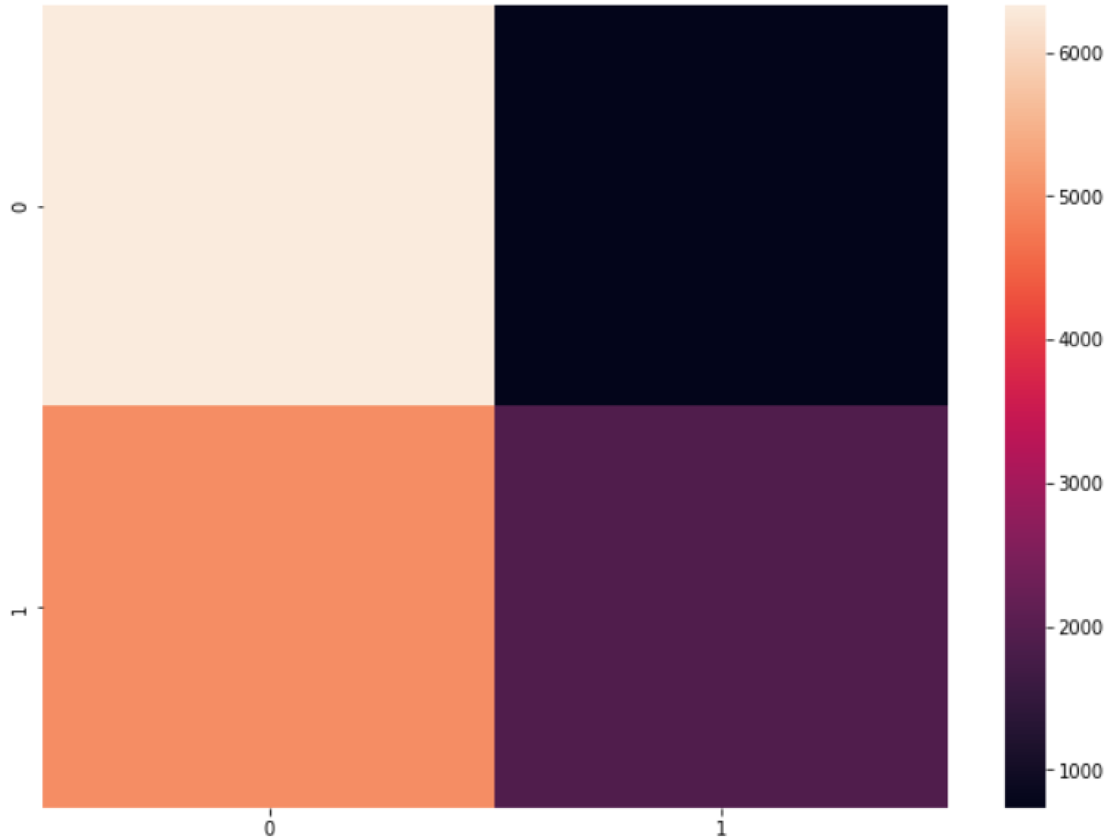

Figure 18 Heat map by using seaborn

CHAPTER 8

ACCURACY

The accuracy achieved through the trained machine learning model is 73.2400%.

```
from sklearn.metrics import accuracy_score, plot_confusion_matrix
y_pred = xgb.predict(X_test)
predictions = [round(value) for value in y_pred]
accuracy = accuracy_score(y_test, predictions)
print("Accuracy: %.4f%%" % (accuracy * 100.0))
```

Figure 19 Accuracy of model

CHAPTER 9

CONCLUSION

I have evaluated two kinds of Machine learning based method for cardiovascular disease, such as K neighbor and Gaussian algorithms. One of the major drawbacks of these works is that the main focus has been on the application of classification techniques for heart disease prediction, rather than studying various data cleaning and pruning techniques that prepare and make a dataset suitable for mining. It has been observed that a properly cleaned and pruned dataset provides much better accuracy than an unclean one with missing values. In the future, we will continue our research to develop new architectures for detecting Cardiovascular disease on large database.

CHAPTER 10

REFERENCE

[1] Shaikh Abdul Hannan, A.V. Mane, R. R. Manza,and R. J. Ramteke, Dec 2010,"Prediction of Heart Disease Medical Prescription using Radial Basis Function",IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), DOI: 10.1109/ICCIC.2010.5705900 ,28-29 .

[2] Mrudula Gudadhe, Kapil Wankhade, and Snehlata Dongre, Sept 2010,"Decision Support System for Heart Disease Based on Support Vector Machine and Artificial Neural Network",International Conference on Computer and Communication Technology (ICCCT),DOI:10.1109/ICCCT.2010.5640377, 17-19.

[3] Asha Rajkumar, and Mrs G. Sophia Reena, 2010, "Diagnosis of Heart Disease using Data Mining Algorithms",Global Journal of Computer Science and Technology,Vol. 10,Issue 10, pp.38-43, September.