1. INTRODUCTION

1.1 Introduction

Machine learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves. The main aim is to allow computers to learn automatically without human intervention and also adjust actions accordingly. Heart disease has a great deal of attention in medical research. The diagnosis of heart disease is a challenging task, which can offer automated prediction about the heart condition of patient so that further treatment can be made effective.

1.2 Existing System

Hospitals maintain all the patient records. Even though, those records are not used in an efficient manner for diagnosis. To maintain the records in an efficient error free manner, the new proposed system is introduced**.**

Disadvantages:

1. Doesn’t generate accurate and efficient results

2. Computation time is very high

3. Difficulty in maintenance of patient records

4. Lacking of accuracy may result in lack of efficient further treatment

1.3 Proposed System

We proposed to develop a system which will help practitioners to predict heart related disease based on some attributes like age, gender, blood pressure and so on. So, there is a need for developing a decision system which will help practitioners to predict the heart disease in an easier way, which can offer prediction about the heart condition of patient so that further treatment can be made effectively. This proposed system not only accurately predicts heart disease but also reduces time for prediction. The machine learning algorithms like decision tree, random forest, Naive Bayes, K Nearest Neighbours have proven to be most accurate & reliable and hence, used in this project.

Advantages:

1. Generates accurate and efficient results

2. Computation time is greatly reduced

3. Easy maintenance of patient records

4. Reduces manual work

5. Efficient further treatment

6. Automated prediction

1.4. System Requirements

1.4.1 Hardware Requirements:

● System Type	:	Intel(R) Core™2 i7-5500U CPU @ 2.40GHz
● Cache memory	:	4MB(Megabyte)
● RAM	:	8 gigabyte (GB)

1.4.2 Software Requirements:

● Operating System	:	Windows 10 Home, 64 bit Operating System
● Coding Language	:	Python
● Python distribution	:	Anaconda, Spyder

2. LITERATURE SURVEY

2.1 Machine Learning

Machine learning is one of the applications of artificial intelligence (AI) that provides computers, the ability to learn automatically and improve from experience instead of explicitly programmed. It focuses on developing computer programs that can access data and use it to learn from themselves. The main aim is to allow computers to learn automatically without human intervention and also adjust actions accordingly.

2.2 Some machine learning methods

Machine learning algorithms are often categorized as supervised and unsupervised.

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
- In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn’t require additional resources.

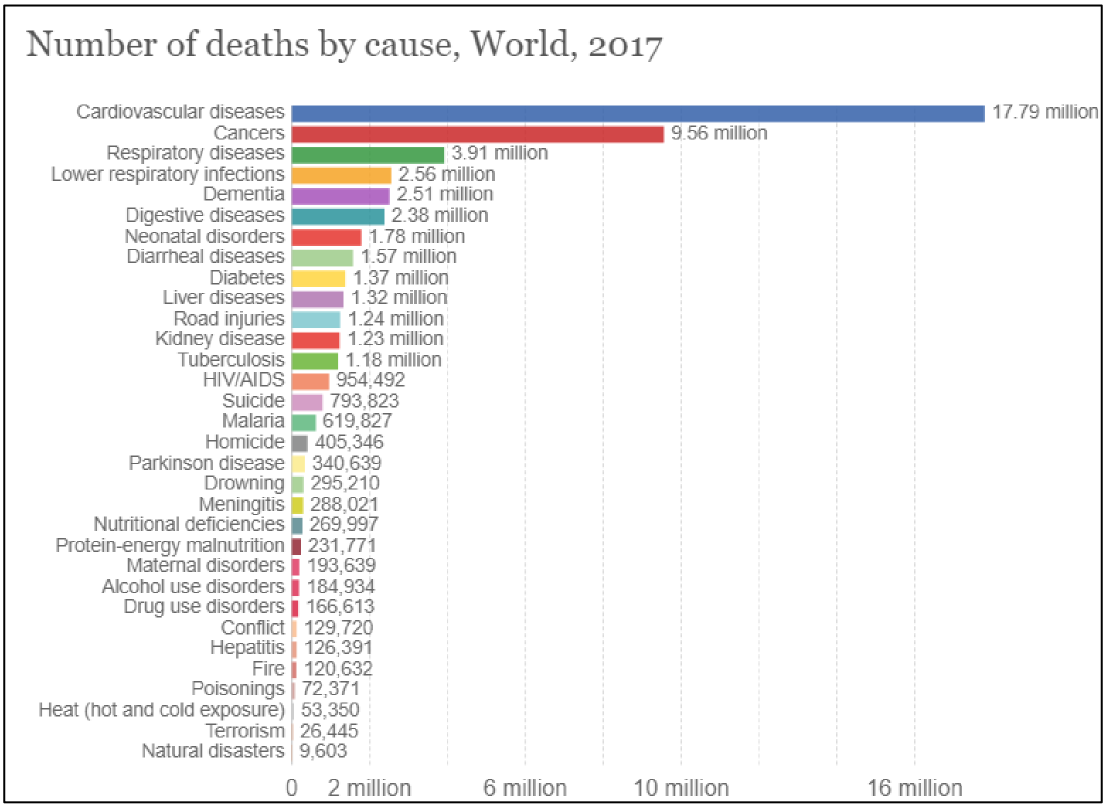
- **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best. This is known as the reinforcement signal.

2.3 Applications of machine learning

1. Virtual Personal Assistants
2. Predictions while Commuting
3. Videos Surveillance
4. Social Media Services
5. Email Spam and Malware Filtering
6. Online Customer Support
7. Search Engine Result Refining
8. Product Recommendations
9. Online Fraud Detection

2.4 Prevalence of cardiovascular diseases

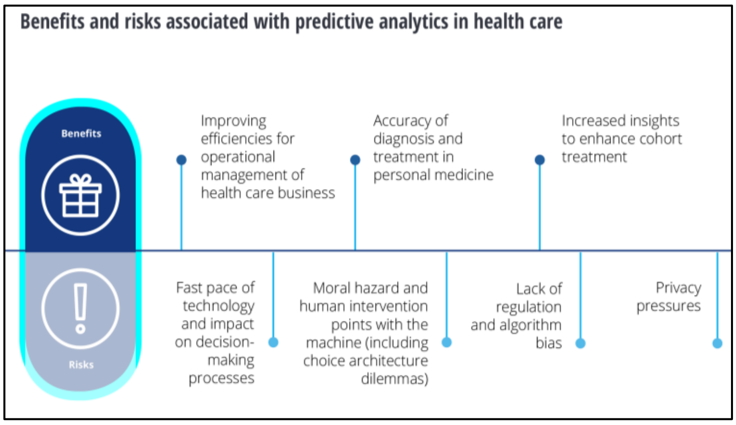
The world health report 2017, there were nearly 57 million deaths in the world in 2015. An estimated 17.79 million deaths occur due to cardiovascular diseases worldwide. More than 75% deaths due to cardiovascular diseases occur in the middle-income and low-income countries. Also, 80% of the deaths that occur due to cardiovascular diseases are because of stroke and heart attack [1]. India too has a growing number of cardiovascular disease patients added every year. Currently, the number of heart disease patients in India is more than 30 million. Over two lakh open heart surgeries are performed in India each year. A matter of growing concern is that the number of patients requiring coronary interventions has been rising at 20% to 30% for the past few years [2].



From the above graph we can say that the death rate due to cardiovascular diseases is more compared to other diseases.

2.5 Importance of machine learning in healthcare

The importance of machine learning in healthcare is increasing because of its ability to process huge datasets efficiently beyond the range of human capability, and then dependably convert analysis of that data into clinical insights that assist physicians in planning and providing care, which ultimately leads to better outcomes, reduces the costs of care, and increases patients satisfaction. Using these types of advanced analytics, we can provide better information to doctors at the point of patient care.


Benefits and risks associated with predictive analytics in health care
Benefits
Improving efficiencies for operational management of health care business
Accuracy of diagnosis and treatment in personal medicine
Increased insights to enhance cohort treatment
Risks
Fast pace of technology and impact on decision-making processes
Moral hazard and human intervention points with the machine (including choice architecture dilemmas)
Lack of regulation and algorithm bias
Privacy pressures

2.6 Implementation of machine learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files.

Python was designed for its readability. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries.

Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

NumPy: NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

SciPy: Scipy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Scikit-learn: Scikit-learn is one of the most popular Machine Learning libraries for classical Machine Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for dat.a-mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

Theano: Theano is a popular python library that is used to define, evaluate and optimize mathematical expressions involving multi-dimensional arrays in an efficient manner. It is achieved by optimizing the utilization of CPU and GPU. It is extensively used for unit-testing and self-verification to detect and diagnose different types of errors. Theano is a very powerful library that has been used in large-scale computationally intensive scientific projects for a long time but is simple and approachable enough to be used by individuals for their own projects.

TensorFlow: TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

Keras: Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best thing about Keras is that it allows for easy and fast prototyping.

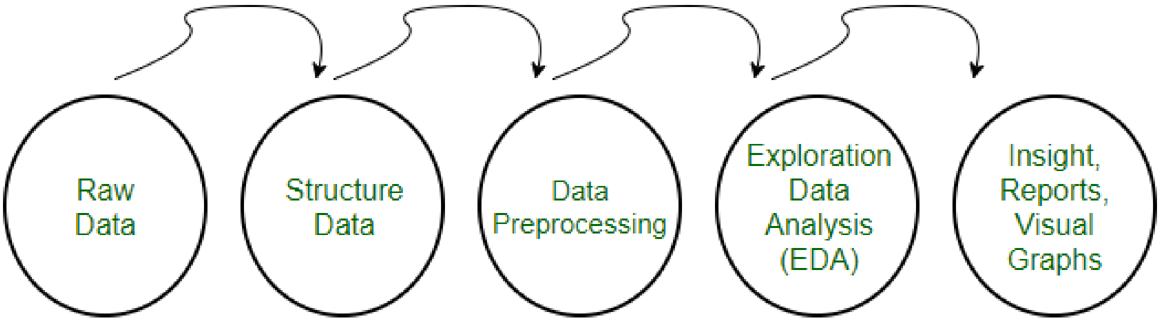
PyTorch: PyTorch is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library which is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that supports on Computer Vision, Natural Language Processing(NLP) and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

Pandas: Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for groping, combining and filtering data.

Matpoltlib: Matpoltlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, histogram, error charts, bar chats, etc.

Data Pre-processing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.


Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

Need of Data Preprocessing

For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format. For example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

Another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in one data set, and best out of them is chosen.

2.7 Classification

- It is a process of categorising data into given classes. Its primary goal is to identify the class of our new data.

2.7.1 Machine learning algorithms for classification

Research on data mining has led to the formulation of several data mining algorithms. These algorithms can be directly used on a dataset for creating some models or to draw vital conclusions and inferences from that dataset. Some popular data mining algorithms are Decision tree, Naïve Bayes, k-means, artificial neural network etc.

1. Decision Tree: Decision Tree Analysis is a general, predictive modelling tool that has applications spanning a number of different areas. In general, decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.The decision rules are generally in form of if-then-else statements. The deeper the tree, the more complex the rules and fitter the model.

2. Naive Bayes (NB): It is a simple technique for constructing classifiers. It is a probabilistic classifier based on Bayes' theorem. All Naive Bayes classifiers assume that the value of any particular feature is independent of the value of any other feature, given the class variable. Bayes theorem is given as follows**: P(C|X) = P(X|C) * P(C)/P(X)**, where X is the data tuple and C is the class such that P(X) is constant for all classes. Though it assumes an unrealistic condition that attribute values are conditionally independent, it performs surprisingly well on large datasets where this condition is assumed and holds.

3. Random Forest: Random Forests are an ensemble learning method (also thought of as a form of nearest neighbour predictor) for classification and regression techniques. It builds multiple decision trees and then merges them together in-order to get more accurate and stable predictions. It constructs a number of Decision trees at training time and outputs the class that is the mode of the classes output by individual trees. It also tries to minimize the problems of high variance and high bias by averaging to find a natural balance between the two extremes. Both R and Python have robust packages to implement this algorithm.

4. KNN: KNN algorithm is one of the simplest classification algorithms and it is one of the most used learning algorithms. KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a dataset in which the data points are separated into several classes to predict the classification of a new sample point. A KNN algorithm uses a data and classifies new data points based on a similarity measures (e.g. distance function, error rate). Classification is done by a majority vote to its neighbours. The data is assigned to the class which has the most nearest neighbours. As we increase the number of nearest neighbours, the value of k, accuracy may increase.

When we say a technique is non-parametric, it means that it does not make any assumptions on the underlying data distribution. In other words, the model structure is determined from the data. If you think about it, it’s pretty useful, because in the “real world”, most of the data does not obey the typical theoretical assumptions made (as in linear regression models, for example). Therefore, KNN could and probably should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution data.

5. Logistic Regression: Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

SMOTE Technique: SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem. It aims to balance class distribution by randomly increasing minority class examples by replicating them.SMOTE synthesises new minority instances between existing minority instances. It generates the virtual training records by linear interpolation for the minority class. These synthetic training records are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. After the oversampling process, the data is reconstructed and several classification models can be applied for the processed data.

2.8 Machine learning products

Data mining is widely used in the medical field such as prediction of heart disease since it is a multidisciplinary field. Using data mining researchers are developing various techniques in-order to predict the heart diseases with high accuracy. Large no. of research work is carried out for medical diagnosis for various diseases

Prediction of heart disease using k-nearest neighbor and particle swarm optimization was introduced Jabber MA[3]. Feature subset selection is used to solve this problem. Feature selection improved accuracy and reduced the running time. Before feature subset selection accuracy obtained is 75%. PSO search filters the number of features and selects the features which contribute more to the classification. By applying KNN with PSO accuracy improved to 100%.

Adaptive weighted fuzzy rule-based system for the risk level assessment of heart disease was proposed by Animesh Kumar Paul Pintu Chandra Shill Md. Rafiqul Islam Rabin Kazuyuki Murase[4]. In this work, a fuzzy system is advanced for the prediction of heart disease’s risk levels using GAs, modified DMSPSO and ensemble technique. This model selects the critical attributes which can assist the heart disease diagnosis. Effective attributes are selected through statistical methods such as Correlation coefficient, R-Squared and Weighted Least Squared (WLS) method, iii) Weighted fuzzy rules are formed on the basis of selected attributes

Prediction of risk score for heart disease using associative classification and hybrid feature subset selection was obtained by Jabbar Akhil[5] used Feature selection as a pre-processing step in used to reduce dimensionality, removing irrelevant data and increasing accuracy and improves comprehensibility. Associative classification is a recent and rewarding technique that applies the methodology of association into classification and achieves high classification accuracy. Most associative classification algorithms adopt exhaustive search algorithms like in Apriori, and generate huge no. of rules from which a set of high quality of rules are chosen to construct efficient classifier.

An Integrated Decision Support System Based on ANN and Fuzzy_AHP for Heart Failure Risk Prediction by Oluwarotimi Williams Samuela, Grace Mojisola Asogbona, Arun Kumar Sangaiahc, Peng Fanga and Guanglin Lia [6]. Fuzzy analytic hierarchy process (Fuzzy_AHP) technique was used to compute the global weights for the attributes based on their individual contribution with an accuracy of 91.10%, which is 4.40% higher in comparison to that of the conventional ANN method.

Ali. Adeli et al., have introduced an expert fuzzy model for the diagnosis of heart disease, the proposed system was analysed on the V.A.medical center, Long beach and Cleveland clinic foundation database ("UCI Machine Learning Repository: Heart Disease Data Set," n.d.). Mamdani inference method is utilized in this model [7] to design the fuzzy expert system (FES) through membership function, fuzzy rule base, fuzzification and defuzzification with 13 inputs and 1 output.

Robert Detrano [8] achieved 77.00% accuracy on Cleveland heart disease data set using logistic regression algorithm. Newton Cheung [9] applied C4.5, Naive Bayes, BNND and BNNF algorithms and obtained 81.11 %, 81.45%, 81.11%, and 80.96% accuracy, respectively on Cleveland data set. WEKA and RA obtained 83.60% accuracy using Naive-Bayes algorithm [10].

Prediction of heart disease using neural network was proposed by Dangare et al. in [11]. Feature selection is used to predict the disease. Their method obtained an accuracy of 92.5% for 13 features and 100% accuracy with 15 features. There is a 7.5% improvement after discarding 2 features from 15 to 13.

Hlaudi Daniel Masethe, Mosima Anna et al. proposed a model using decision tree for heart disease prediction. Authors compared their approach with other classification approaches. RepTree and J48 achieved an accuracy of 99.07%.

3. SYSTEM ANALYSIS

3.1 Scope of the project

The scope of this system is to maintain patient details in datasets, train the model using the large quantity of data present in datasets and predict whether presence or absence of disease on new data during testing.

3.2 Analysis

The dataset used in this work is Cleveland dataset which is obtained from UCI ('University of California, Irvine') Machine Learning repository.

The dataset contains 14 attributes which are used to predict the heart disease such as

1. age: in years
2. sex
3. cp: chest pain type
4. trestbps: resting blood pressure
5. chol: serum cholestoral in mg/dl
6. fbs: (fasting blood sugar > 120 mg/dl)
7. restecg: resting electrocardiographic results
8. thalach: maximum heart rate achieved
9. exang: exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. slope: the slope of the peak exercise ST segment
12. ca: number of major vessels (0-3) colored by flourosopy
13. thal
14. num: diagnosis of heart disease (angiographic disease status)

    It is integer valued from 0 (no presence) to 4.

    The dataset contains 303 instances and 5 classes i.e., 0,1,2,3,4.

The dataset is converted into a Microsoft excel file.

Age	Sex	cp	trestbps	chol	fbs	restecg	thalach	exang	Oldpeak	slope	ca	thal	Num
63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
48	1	2	130	245	0	2	180	0	0.2	2	0	3	0
58	1	4	150	270	0	2	111	1	0.8	1	0	7	3
45	1	4	104	208	0	2	148	1	3	2	0	3	0
62	1	3	130	231	0	0	146	0	1.8	2	3	7	0
44	0	3	108	141	0	0	175	0	0.6	2	0	3	0
63	0	3	135	252	0	2	172	0	0	1	0	3	0
52	1	4	128	255	0	0	161	1	0	1	1	7	1
59	1	4	110	239	0	2	142	1	1.2	2	1	7	2
60	0	4	150	258	0	2	157	0	2.6	2	2	7	3
52	1	2	134	201	0	0	158	0	0.8	1	1	3	0
48	1	4	122	222	0	2	186	0	0	1	0	3	0
45	1	4	115	260	0	2	185	0	0	1	0	3	0
34	1	1	118	182	0	2	174	0	0	1	0	3	0
57	0	4	128	303	0	2	159	0	0	1	1	3	0
71	0	3	110	265	1	2	130	0	0	1	1	3	0
49	1	3	120	188	0	0	139	0	2	2	3	7	3
54	1	2	108	309	0	0	156	0	0	1	0	7	0
59	1	4	140	177	0	0	162	1	0	1	1	7	2
57	1	3	128	229	0	2	150	0	0.4	2	1	7	1
61	1	4	120	260	0	0	140	1	3.6	2	1	7	2
39	1	4	118	219	0	0	140	0	1.2	2	0	7	3
61	0	4	145	307	0	2	146	1	1	2	0	7	1
56	1	4	125	249	1	2	144	1	1.2	2	1	3	1
52	1	1	118	186	0	2	190	0	0	2	0	6	0
43	0	4	132	341	1	2	136	1	3	2	0	7	2
62	0	3	130	263	0	0	97	0	1.2	2	1	7	2

3.3 Data Pre-processing

Before feeding data to an algorithm we have to apply transformations to our data which is referred as pre-processing. By performing pre-processing the raw data which is not feasible for analysis is converted into clean data. In-order to achieve better results using a model in Machine Learning, data format has to be in a proper manner. The data should be in a particular format for different algorithms. For example, if we consider Random Forest algorithm it does not support null values. So that those null values have to be managed using raw data.

3.3.1 Missing values

Filling missing values is one of the pre-processing techniques. The missing values in the dataset is represented as '?' but it a non-standard missing value and it has to be converted into a standard missing value NaN. So that pandas can detect the missing values. The fig1 below is a heatmap representing the missing values. In these graph missing values are present in ca, thal features. We have filled that missing values using the median of the features. After filling the missing values our heat map looks like fig2.

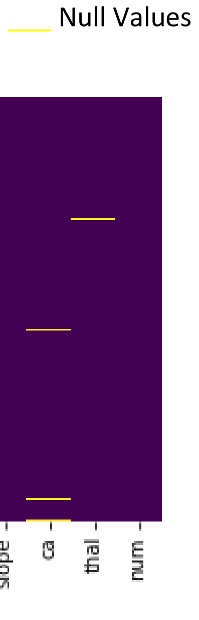


fig:3.3.1.1 Missing data visualisation

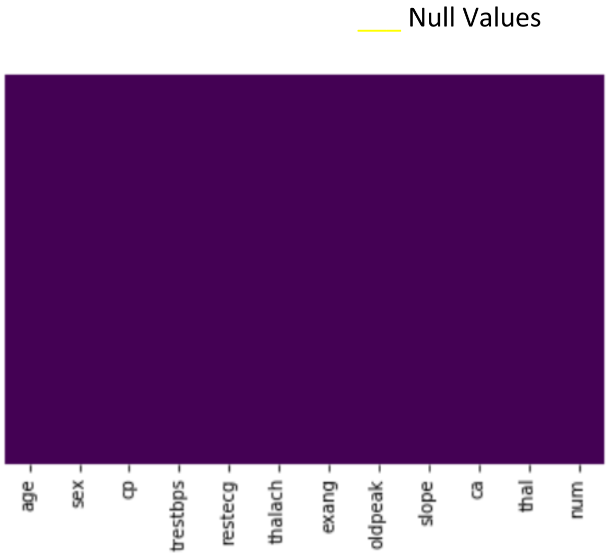
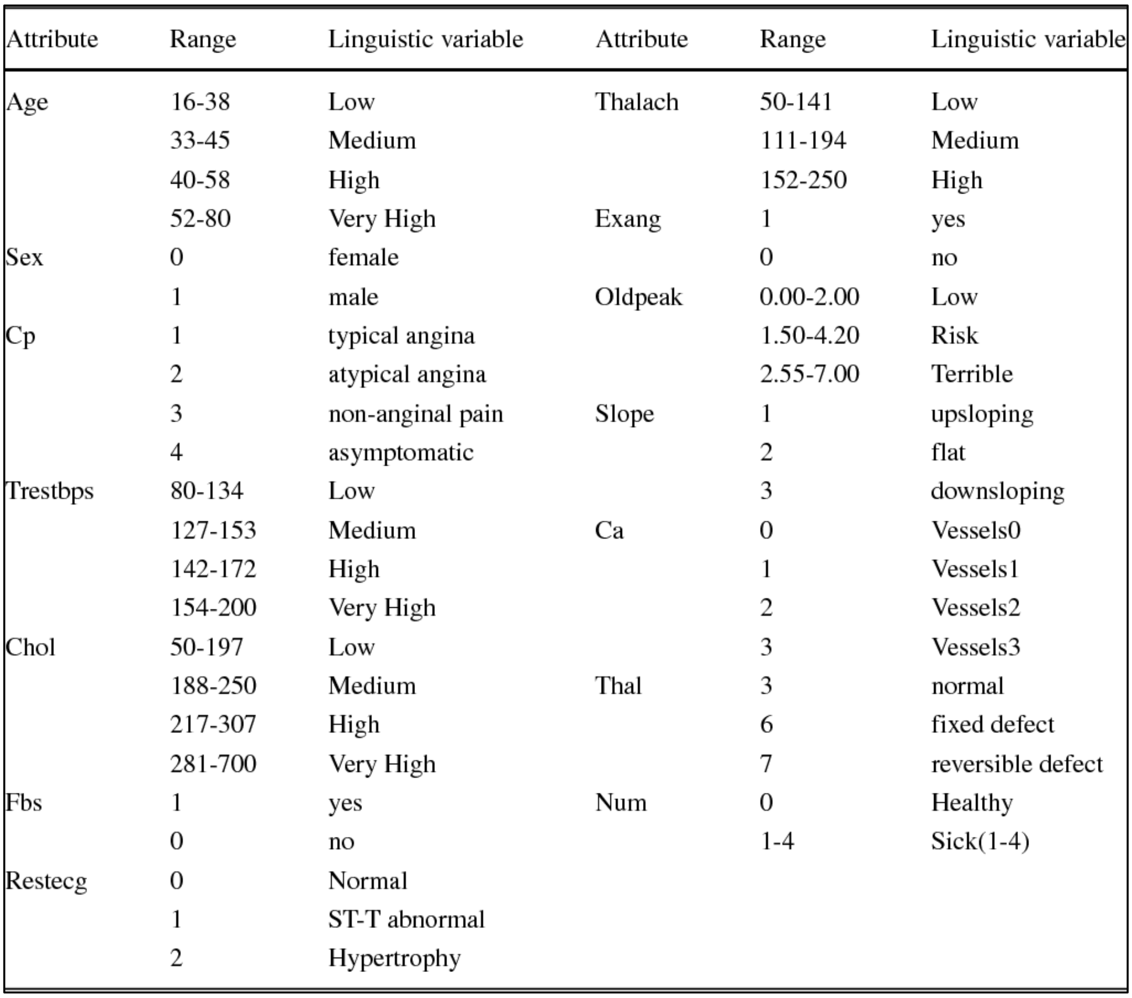


fig:3.3.1.2 Missing data visualisation After preprocessing

3.3.2 Discretization

Conversion of a continuous-valued variable to a discrete value by creating a set of contiguous intervals that falls in the range of the variable’s values is called discretization. We have categorised the using the following table. A classification algorithm such as Random Forests (RF) is chosen for its ability to handle high-dimensional data, benefits from discretization in the analysis of genetic data. Decision Tree is able to deal with both continuous and categorical attributes for classification. However, this technique handles the continuous attributes by discretization. A simple probabilistic classification algorithm i.e., Naive Bayes (NB) benefits from discretization of data in-order to perform well in many domains. But when we are using knn since our data contains both continuous and categorical attributes we should use dummies.



Attribute	Range	Linguistic variable	Attribute	Range	Linguistic variable
Age	16-38	Low	Thalach	50-141	Low
	33-45	Medium		111-194	Medium
	40-58	High		152-250	High
	52-80	Very High	Exang	1	yes
Sex	0	female		0	no
	1	male	Oldpeak	0.00-2.00	Low
Cp	1	typical angina		1.50-4.20	Risk
	2	atypical angina		2.55-7.00	Terrible
	3	non-anginal pain	Slope	1	upsloping
	4	asymptomatic		2	flat
Trestbps	80-134	Low		3	downsloping
	127-153	Medium	Ca	0	Vessels0
	142-172	High		1	Vessels1
	154-200	Very High		2	Vessels2
Chol	50-197	Low		3	Vessels3
	188-250	Medium	Thal	3	normal
	217-307	High		6	fixed defect
	281-700	Very High		7	reversible defect
Fbs	1	yes	Num	0	Healthy
	0	no		1-4	Sick(1-4)
Restecg	0	Normal			
	1	ST-T abnormal			
	2	Hypertrophy			

If we consider the class labels class label 0 is in 164 . class label 1 is in 55 instances . class label 2 is in 36 instances. class label 3 is in 35 instances. class label 4 is in 13 instances.This dataset is suffering from class imbalence.So we have balenced the class labels by transforming 1,2,3,4 class labels as 1.

3.3.3 Feature scaling

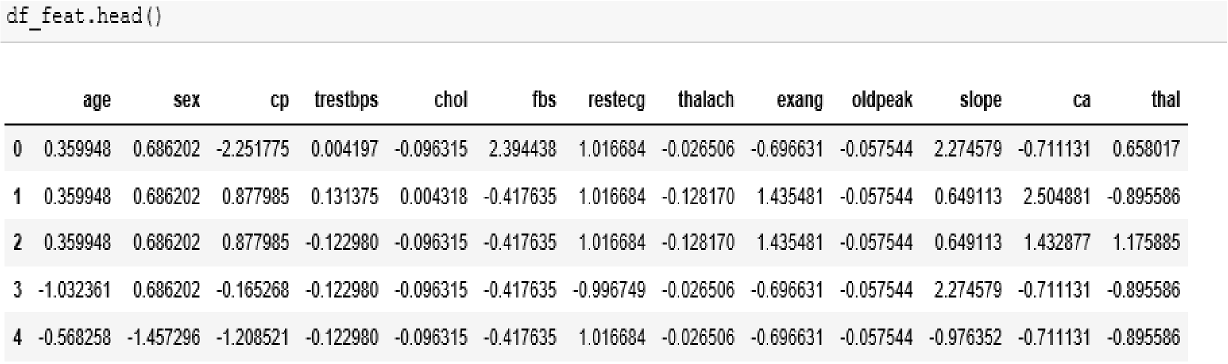
It is nothing but data normalization in data processing used to standardize the range of independent variables. This feature is very useful in data pre-processing step.

The Standard Scaler will assume that the data is normally distributed within each attribute and will scale them in such a way that the distribution is now centred on 0, with a standard deviation is of 1.The mean and standard deviation are calculated for the feature and then the feature is scaled based on:

y_i –mean(y)/stdev(y)

y_i represents the values of attribute y

After scaling data the data set looks like



```
df_feat.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	0.359948	0.686202	-2.251775	0.004197	-0.096315	2.394438	1.016684	-0.026506	-0.696631	-0.057544	2.274579	-0.711131	0.658017
1	0.359948	0.686202	0.877985	0.131375	0.004318	-0.417635	1.016684	-0.128170	1.435481	-0.057544	0.649113	2.504881	-0.895586
2	0.359948	0.686202	0.877985	-0.122980	-0.096315	-0.417635	1.016684	-0.128170	1.435481	-0.057544	0.649113	1.432877	1.175885
3	-1.032361	0.686202	-0.165268	-0.122980	-0.096315	-0.417635	-0.996749	-0.026506	-0.696631	-0.057544	2.274579	-0.711131	-0.895586
4	-0.568258	-1.457296	-1.208521	-0.122980	-0.096315	-0.417635	1.016684	-0.026506	-0.696631	-0.057544	-0.976352	-0.711131	-0.895586

3.3.4 Correlation coefficient method

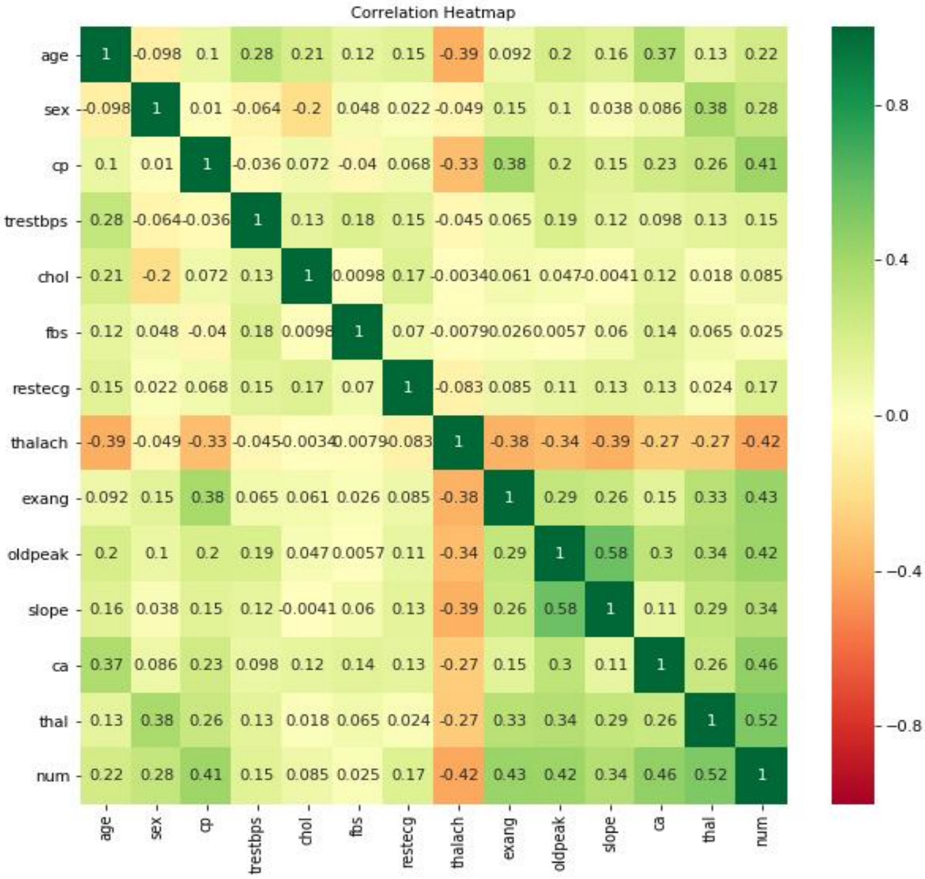
We can find dependency between two attributes p and q using Correlation coefficient method using the formula.

$$r_{p,\,q}= \sum(p_i-\overline{p})(q_i-\overline{q})/n\sigma_p\sigma_q$$

$$=\sum(p_i\ q_i)-n\overline{p}\ \overline{q}/\ n\sigma_p\sigma_q$$

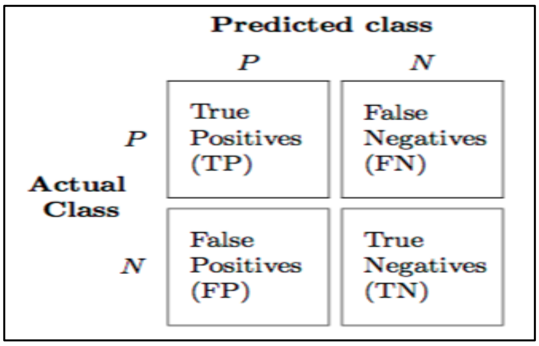
n is the total number of patterns, p_i and q_i are respective values of p and q attributes in patterns i, $\overline{p}$ and $\overline{q}$ are respective mean values of p and q attributes, σ_p , σ_q are respective standard deviations values of p and q attributes. Generally, $-1\leq r_{p,q} \leq +1$. If $r_{p,q} < 0$, then p and q are negatively correlated. If $r_{p,q} =0$, then p and q are independent attributes and there is no correlation between them. If $r_{p,q} > 0$, then p and q are positively correlated.

We can drop the attributes that are having correlation coefficient value as 0 as it indicates that the variables are independent with respect to the prediction attribute.



3.4 CONFUSION MATRIX

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.



		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

A true positive (tp) is a result where the model predicts the positive class correctly. Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class.

A false positive (fp) is an outcome where the model incorrectly predicts the positive class. And a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

Sensitivity or Recall or hit rate or true positive rate (TPR)

It is the proportion of individuals who actually have the disease were identified as having the disease.

TPR = tp / (tp + fn)

Specificity, selectivity or true negative rate (TNR)

It is the proportion of individuals who actually do not have the disease were identified as not having the disease.

TNR = tn / (tn + fp) =1-FPR

Precision or positive predictive value (PPV)

If the test result is positive what is the probability that the patient actually has the disease.

PPV = tp / (tp + fp)

Negative predictive value (NPV)

If the test result is negative what is the probability that the patient does not have disease.

NPV = tn / (tn + fn)

Miss rate or false negative rate (FNR)

It is the proportion of the individuals with a known positive condition for which the test result is negative.

FNR = fn / (fp + tn)

Fall-out or false positive rate (FPR)

It is the proportion of all the people who do not have the disease who will be identified as having the disease.

FPR = fp/ (fp + tn)

False discovery rate (FDR)

It is the proportion of all the people identified as having the disease who do not have the disease.

FDR = fp / fp + tp

False omission rate (FOR)

It is the proportion of the individuals with a negative test result for which the true condition is positive.

FOR = fn / (fn + tn)

Accuracy

The accuracy reflects the total proportion of individuals that are correctly classified.

ACC = (tp + tn) / (tp + tn + fp + fn)

F1 score

It is the harmonic mean of precision and sensitivity

F1 = 2tp / (2tp+ fp + fn)

4. IMPLEMENTATION CODE

Logistic Regression using SMOTE Technique

```
import pandas as pd

from sklearn.model_selection import train_test_split # used for splitting training and testing data

from sklearn.preprocessing import StandardScaler # used for feature scaling

from sklearn.linear_model import LogisticRegression

import pickle

dataset1 = pd.read_csv('cleveland14.csv')

dataset1.fillna(dataset1.mean(), inplace=True)

C = dataset1.iloc[:, :-1].values # attributes to determine dependent variable / Class

D = dataset1.iloc[:, -1].values# dependent variable / Class

X_train1, X_test1, y_train1, y_test1 = train_test_split(C,D, test_size = 0.23, random_state = 46)

sc_X1 = StandardScaler()

X_train1 = sc_X1.fit_transform(X_train1)

X_test1 = sc_X1.transform(X_test1)

from imblearn.over_sampling import SMOTE

sm = SMOTE(random_state = 2)

X_train_res, y_train_res = sm.fit_sample(X_train1, y_train1.ravel())

LR1=LogisticRegression()

# Train the model

LR1=LR1.fit(X_train_res, y_train_res)

# Saving model to disk

pickle.dump(LR1, open('model.pkl','wb'))
```

```
# Loading model to compare the results

model = pickle.load(open('model.pkl','rb'))
```

Flask Code to connect frontEnd

```
import numpy as np

from flask import Flask,request, jsonify, render_template

import pickle

app = Flask(__name__)

model = pickle.load(open('model.pkl', 'rb'))

@app.route("/")

def home():

    return render_template("index.html")

@app.route("/form")

def form():

    return render_template("form.html")

@app.route("/output")

def output():

    return render_template("output.html")

@app.route("/risk")

def risk():

    return render_template("risk.html")

@app.route("/contact")

def contact():

    return render_template("contact.html")

@app.route('/predict',methods=['POST'])

def predict():
```

```
int_features = []
age=request.form['age']
int_features.append(int(age))
gender=request.form['gender']
int_features.append(int(gender))
cpt=request.form['cpt']
int_features.append(int(cpt))
slider=request.form['slider']
int_features.append(int(slider))
sc=request.form['sc']
int_features.append(int(sc))
fbs=request.form['fbs']
int_features.append(int(fbs))
rer=request.form['rer']
int_features.append(int(rer))
mh=request.form['mh']
int_features.append(int(mh))
eia=request.form['eia']
int_features.append(int(eia))
oldpeak=request.form['oldpeak']
int_features.append(float(oldpeak))
slope=request.form['slope']
int_features.append(int(slope))
ca=request.form['ca']
int_features.append(int(ca))
```

```
    thala=request.form['thala']
    int_features.append(int(thala))
    print(int_features)
    final_features = [np.array(int_features)]
    print(final_features)
    prediction = model.predict(final_features)
    print(prediction)
    output = '{0:.{1}f}'.format(prediction[0],2)
    if output<str(0.5):
        return render_template('output.html', prediction='Diagnosis suggest that you have no heart disease.')
    else:
        return render_template('output.html', prediction='Diagnosis suggest that you have heart disease')
if __name__ == "__main__":
    app.run(debug=True)
```

Form.html

```
<!doctype html>
<html lang="en">
<head>
<!-- Required meta tags -->
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@9"></script>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="icon" href="img/favicon.png" type="image/png">
```

```
<title>Hospice Medical</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.css') }}">
<link rel="stylesheet" href="{{ url_for('static', filename='vendors/linericon/style.css') }}">
<link rel="stylesheet" href="{{ url_for('static', filename='css/font-awesome.min.css') }}">
<link rel="stylesheet" href="{{ url_for('static', filename='vendors/owl-carousel/owl.carousel.min.css') }}">
<link rel="stylesheet" href="{{ url_for('static', filename='vendors/lightbox/simpleLightbox.css') }}">
<link rel="stylesheet" href="{{ url_for('static', filename='vendors/nice-select/css/nice-select.css') }}">
<link rel="stylesheet" href="{{ url_for('static', filename='vendors/animate-css/animate.css') }}">
<link rel="stylesheet" href="{{ url_for('static', filename='vendors/jquery-ui/jquery-ui.css') }}">
<!-- main css -->
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
<link rel="stylesheet" href="{{ url_for('static', filename='css/responsive.css') }}">
<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
</head>
<body>
<!--================Header Menu Area =================-->
<header class="header_area">
<div class="top_menu row m0">
<div class="container">
```

```
<div class="float-left">

<ul class="left_side">

<li>

<a href="#">

<i class="fa fa-facebook-f"></i>

</a>

</li>

<li>

<a href="#">

<i class="fa fa-twitter"></i>

</a>

</li>

<li>

<a href="#">

<i class="fa fa-dribbble"></i>

</a>

</li>

<li>

<a href="#">

<i class="fa fa-behance"></i>

</a>

</li>

</ul>

</div>

<div class="float-right">
```

```
<ul class="right_side">
<li>
<a href="#">
<i class="lnr lnr-phone-handset"></i>
7330959074
</a>
</li>
<li>
<a href="#">
<i class="lnr lnr-envelope"></i>
chandranimadhira@gmail.com
</a>
</li>
</ul>
</div>
</div>
</div>
<div class="main_menu">
<nav class="navbar navbar-expand-lg navbar-light" style="left: 0px; top: 0px; height: 80px">
<div class="container">
<img src="{{ url_for('static',    filename='img/logo.png') }}" alt=""><!-- Collect the nav links, forms, and other content for toggling --></div>
<div class="collapse navbar-collapse offset" id="navbarSupportedContent">
<div class="row ml-0 w-100">
```

```
<div class="col-lg-12 pr-0" style="height: 10px;">

<ul class="nav navbar-nav center_nav pull-right" style="width: 600px;">

<li class="nav-item">

<a class="nav-link active" href="{{ url_for('home') }}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url_for('risk') }}">Risk Predictor</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url_for('form') }}">Heart Disease Predictor</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url_for('contact') }}">Contact Us</a>

</li>

</ul>

</div>

</div>

</div>

</div>

</nav>

</div>

</header>

<section class="banner_area">

<div class="banner_inner d-flex align-items-center">

<div class="container">
```

```
<div class="banner_content text-left">
</div>
</div>
</div>
</section>
<section class="appointment-area" style="height:900px;">
<div class="container">
<div class="row justify-content-between align-items-center appointment-wrap">
<div class="col-lg-5 col-md-6 appointment-left">
<ul class="time-list">
<li class="d-flex justify-content-between">
<span></span>
</li>
<li class="d-flex justify-content-between">
<span><img src="{{ url_for('static',    filename='img/heart.jpg') }}" alt=""
class="auto-style1" style="height:550px;"></span>
</li>
</ul>

</div>
<div class="col-lg-6 col-md-6 pt-60 pb-60">
<div class="appointment-right">
<form class="form-wrap" method="post" action="{{ url_for('predict')}}">
<h3 class="pb-20 text-center mb-20">Check Your Heart Care Results</h3>
```

```
<input type="number" id="age" class="form-control" min="1" max="100"
name="age" placeholder="Patient Age" required>
<div class="form-select" id="service-select" >
<select name="gender">
<option value="" disabled="disabled" selected>Gender</option>
<option value="0">Female</option>
<option value="1">Male</option>
</select>
</div>
<div class="form-select" id="service-select" >
<select name="cpt">
<option value="" disabled="disabled" selected>Chest Pain Type</option>
<option value="1">typical angina</option>
<option value="2">atypical angina</option>
<option value="3">non-anginal pain</option>
<option value="4">asymptomatic</option>
</select>
</div>
<div><p>Resting Bood Pressure</p></div>
<div class="slidecontainer">
<input type="range" min="0" max="290" value="120" name="slider" class="slider"
id="myRange">
<p><b>Value:</b><span id="demo"></span></p>
</div>
<input type="number" min="1" class="form-control" name="sc" id="sc"
placeholder="Serum Cholestrol" required>
```

```
<div class="form-select" id="service-select" >

<select name="fbs">

<option value="" disabled="disabled" selected>Fasting blood sugar > 120
mg/dl</option>

<option value="0">No</option>

<option value="1">Yes</option>

</select>

</div>

<div class="form-select" id="service-select" >

<select name="rer">

<option value="" disabled="disabled" selected>Resting electrocardiographic
results</option>

<option value="0">Normal</option>

<option value="1">having ST-T wave abnormality</option>

<option value="2">showing probable or definite left ventricular hypertrophy by Estes'
criteria</option>

</select>

</div>

<input type="number" min="1"  class="form-control" id="mh" name="mh"
placeholder="Maximum heart rate achieved" required>

<div class="form-select" id="service-select" >

<select name="eia">

<option value="" disabled="disabled" selected>Exercise induced angina</option>

<option value="0">No</option>

<option value="1">Yes</option>

</select>
```

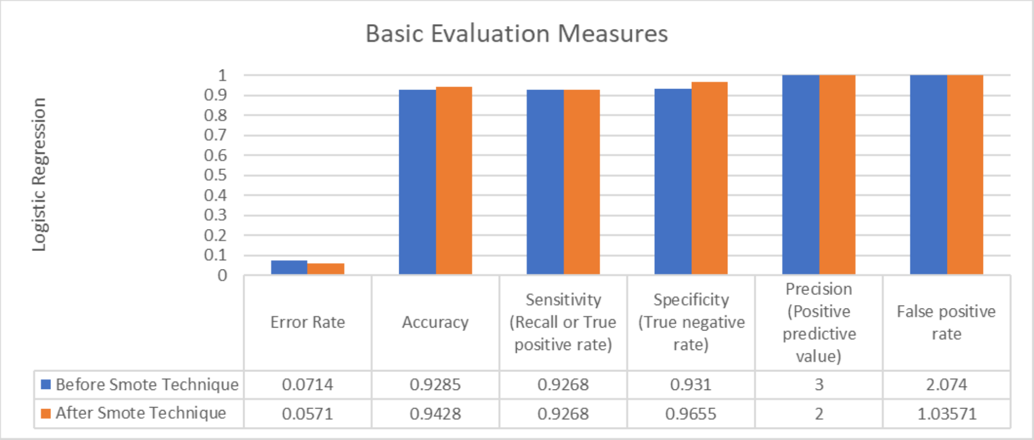
```
</div>
<input type="number" min="0"  class="form-control" step="0.01" id="oldpeak" name="oldpeak" placeholder="ST depression induced by exercise relative to rest" required>
<div class="form-select" id="service-select">
<select name="slope">
<option value="" disabled="disabled" selected>Slope of the peak exercise ST segment</option>
<option value="1">Upsloping</option>
<option value="2">Flat</option>
<option value="3">Downsloping</option>
</select>
</div>
<input type="number" min="0" max="3" class="form-control" id="ca" name="ca" placeholder="Number of major vessels (0-3) colored by flourosopy" required>
<div class="form-select" id="service-select">
<select name="thala">
<option value="" disabled="disabled" selected>Thalassemia</option>
<option value="3">Normal</option>
<option value="6">Fixed defect</option>
<option value="7">Reversable defect</option>
</select>
</div>
<div>
<div class="text-center">
<button class="main_btn text-uppercase">Submit</button>
```

```
</div>
</form>


<!-- End footer Area -->
</div>
</div>
</div>
</div>
<!---          <div class="popup" id="my_dialog"> <a href="{{ url_for('form') }}"
class="close"><svg version="1.1" id="Layer_1"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
x="0px" y="0px" width="10px" height="10px" viewBox="215.186 215.671 80.802
80.8"
enable-background="new 215.186 215.671 80.802 80.8" xml:space="preserve">
<polygon fill="#FFFFFF" points="280.486,296.466 255.586,271.566
230.686,296.471 215.19,280.964 240.086,256.066 215.186,231.17
230.69,215.674 255.586,240.566 280.475,215.671 295.985,231.169 271.089,256.064
295.987,280.96 "
/>
</svg></a>


<div class="valid">
<svg version="1.1" id="Layer_2" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
x="0px" y="0px" width="15px" height="15px" viewBox="222.744 227.408 67.744
58.526"
enable-background="new 222.744 227.408 67.744 58.526" xml:space="preserve">
```

```
<path fill="#39BA6F" d="M250.062,285.934c-9.435-11.111-15.731-18.195-27.318-
28.935l5.793-5.357
c6.778,3.28,11.076,5.774,18.693,11.204c14.32-16.25,23.783-24.495,41.372-
35.438l1.886,4.335
C275.983,244.402,265.359,258.502,250.062,285.934z" />
</svg>
</div>
<h1>Your Predicted Output</h1>
<div class="auto-style5">
<p class="start" id="para"><b>{{prediction}}</b></p>
</div>
</div>  -->
</section>
<footer class="footer-area section_gap">
</footer>
<script src="{{ url_for('static',    filename='js/jquery-3.2.1.min.js') }}"></script>
<script src="{{ url_for('static',    filename='js/popper.js') }}"></script>
<script src="{{ url_for('static',    filename='js/bootstrap.min.js') }}"></script>
<script src="{{ url_for('static',    filename='vendors/nice-select/js/jquery.nice-
select.min.js') }}"></script>
<script src="{{ url_for('static',    filename='vendors/owl-carousel/owl.carousel.min.js')
}}"></script>
<script src="{{ url_for('static',    filename='js/jquery.ajaxchimp.min.js') }}"></script>
<script src="{{ url_for('static',
filename='https://cdnjs.cloudflare.com/ajax/libs/Counter-
Up/1.0.0/jquery.counterup.min.js') }}"></script>
```

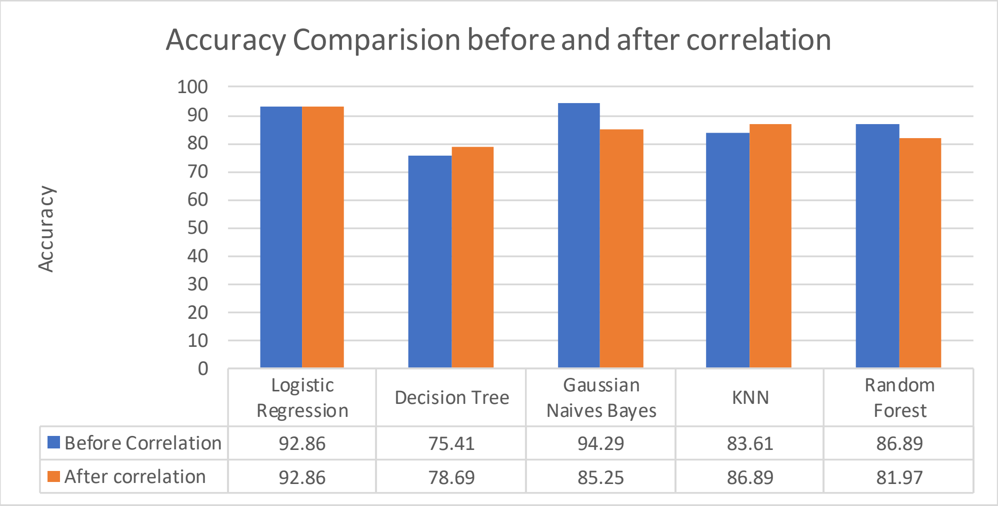
```
<script src="{{ url_for('static',
filename='https://cdnjs.cloudflare.com/ajax/libs/waypoints/4.0.1/jquery.waypoints.min.js') }}"></script>
<script src="{{ url_for('static',    filename='js/mail-script.js') }}"></script>
<script src="{{ url_for('static',    filename='js/custom.js')}}"></script>
<script>
var slider = document.getElementById("myRange");
var output = document.getElementById("demo");
output.innerHTML = slider.value;
slider.oninput = function() {
output.innerHTML = this.value;
}
</script>
</body>
</html>
```

5. RESULT ANALYSIS



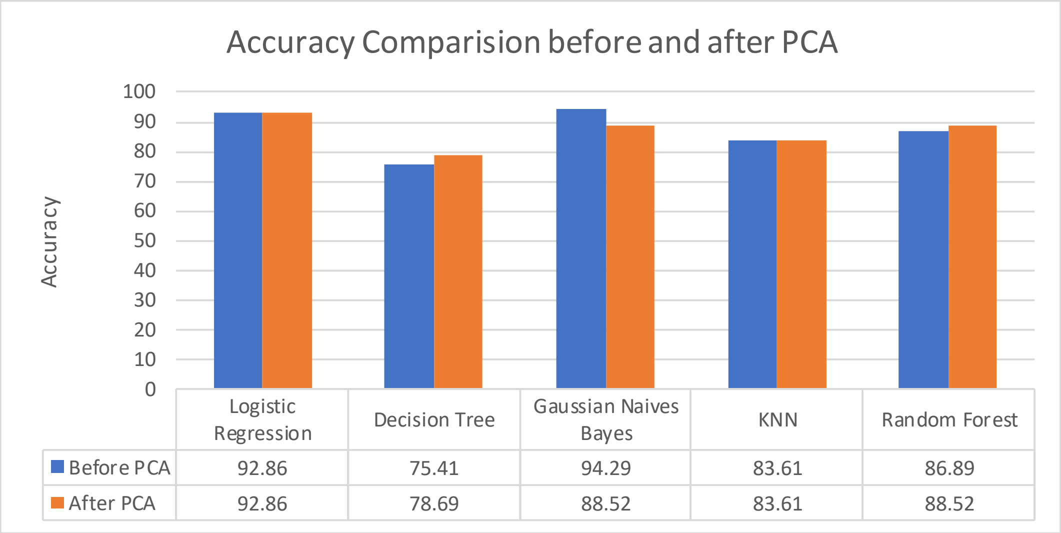
	Error Rate	Accuracy	Sensitivity (Recall or True positive rate)	Specificity (True negative rate)	Precision (Positive predictive value)	False positive rate
🟦 Before Smote Technique	0.0714	0.9285	0.9268	0.931	3	2.074
🟧 After Smote Technique	0.0571	0.9428	0.9268	0.9655	2	1.03571

Fig 3.6.1: comparison of parameters of logistic regression before and after SMOTE Technique



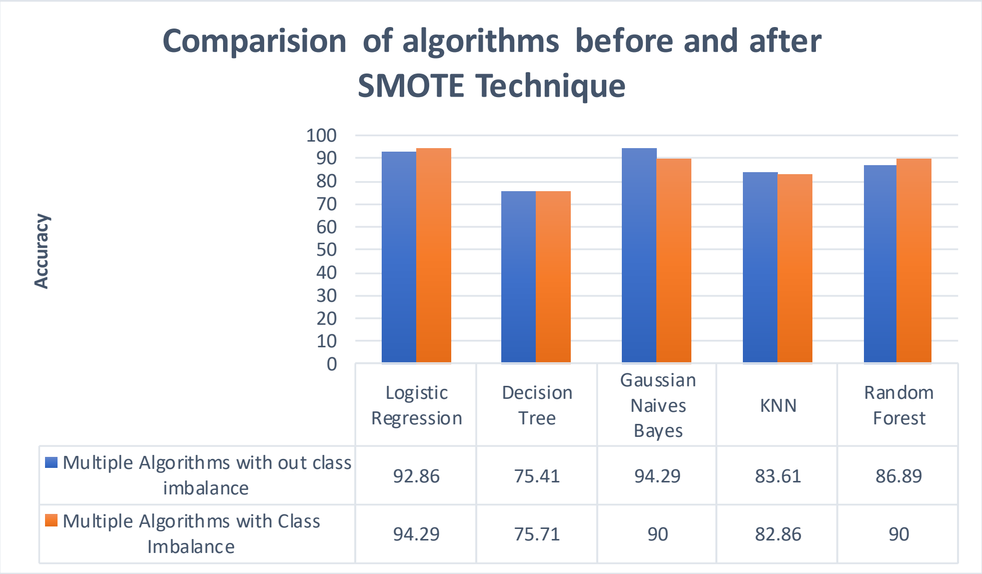
	Logistic Regression	Decision Tree	Gaussian Naives Bayes	KNN	Random Forest
🟦 Before Correlation	92.86	75.41	94.29	83.61	86.89
🟧 After correlation	92.86	78.69	85.25	86.89	81.97

Fig 3.6.2: comparison of algorithms before and after Correlation



	Logistic Regression	Decision Tree	Gaussian Naives Bayes	KNN	Random Forest
🟦 Before PCA	92.86	75.41	94.29	83.61	86.89
🟧 After PCA	92.86	78.69	88.52	83.61	88.52

Fig 3.6.3: Accuracy comparision of algorithms before and after PCA

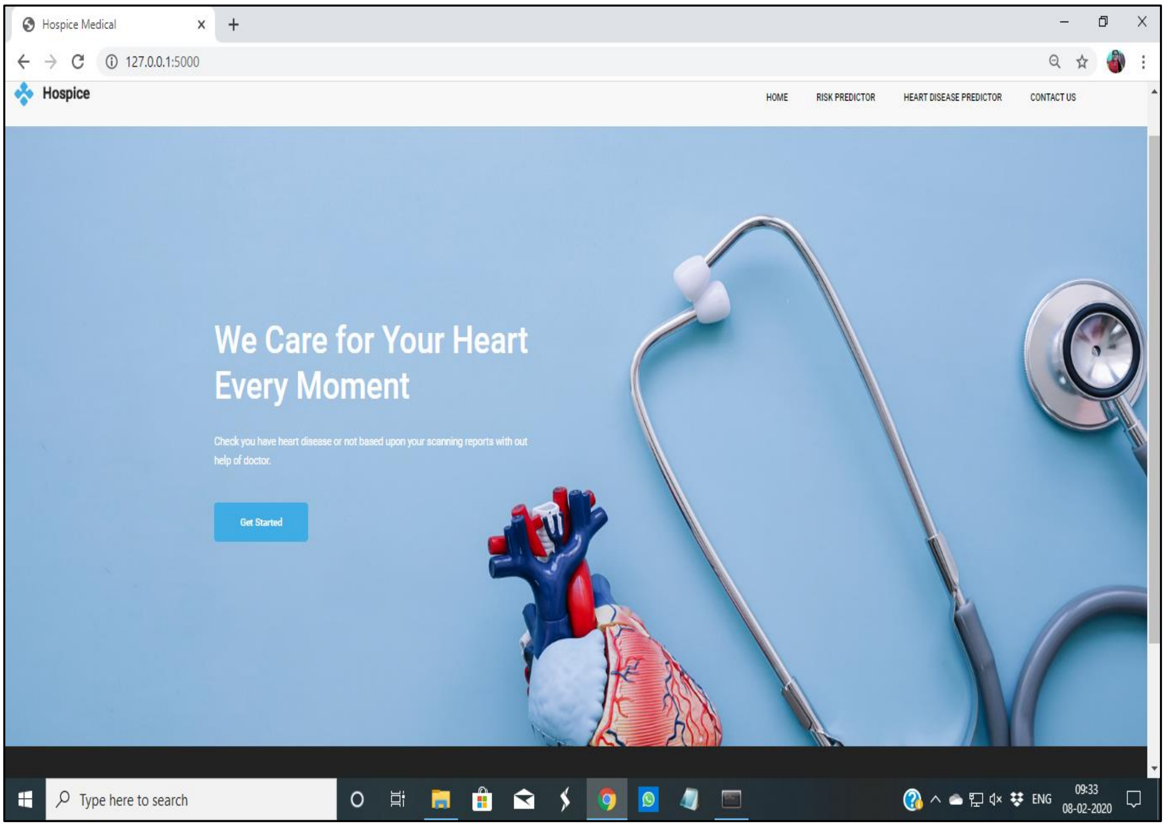


	Logistic Regression	Decision Tree	Gaussian Naives Bayes	KNN	Random Forest
🟦 Multiple Algorithms with out class imbalance	92.86	75.41	94.29	83.61	86.89
🟧 Multiple Algorithms with Class Imbalance	94.29	75.71	90	82.86	90

Fig 3.6.4: comparison of algorithms before and after SMOTE Technique

6.SCREEN SHOTS

HOME PAGE

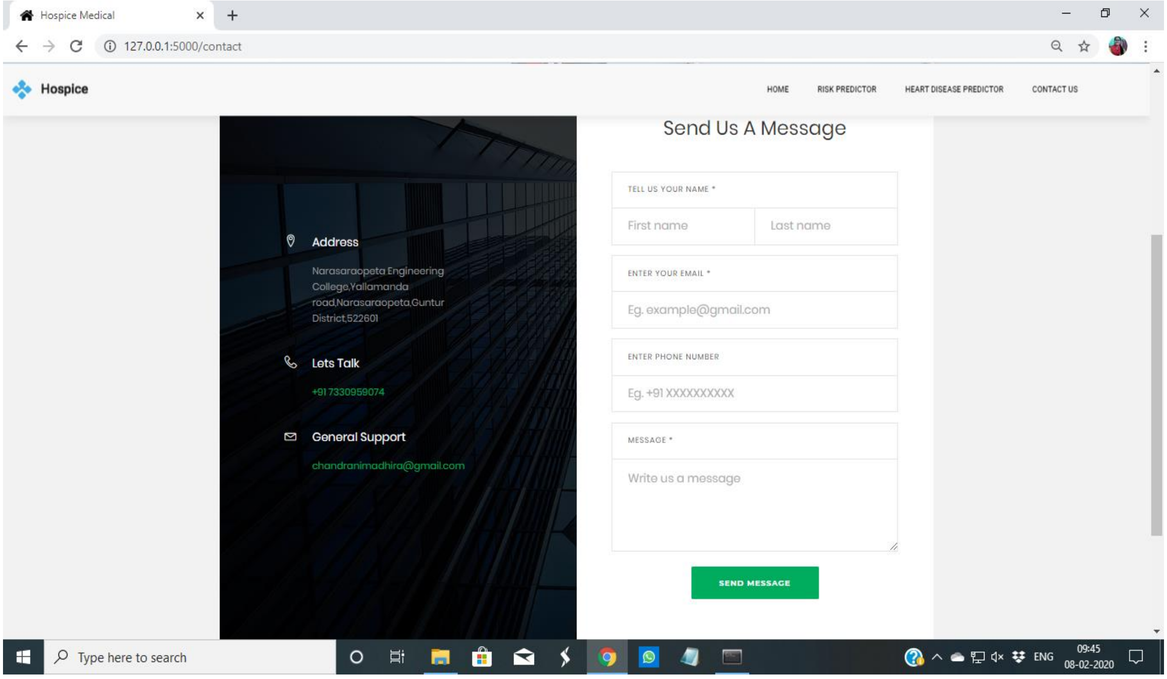


We Care for Your Heart Every Moment

Check you have heart disease or not based upon your scanning reports with out help of doctor.

Fig.4.1 Output screen for Home page

CONTACT US PAGE



Address

Narasaraopeta Engineering College,Yallamanda road,Narasaraopeta,Guntur District,522601

Lets Talk

+91 7330959074

General Support

chandranimadhira@gmail.com

Send Us A Message

TELL US YOUR NAME *

First name

Last name

ENTER YOUR EMAIL *

Eg. example@gmail.com

ENTER PHONE NUMBER

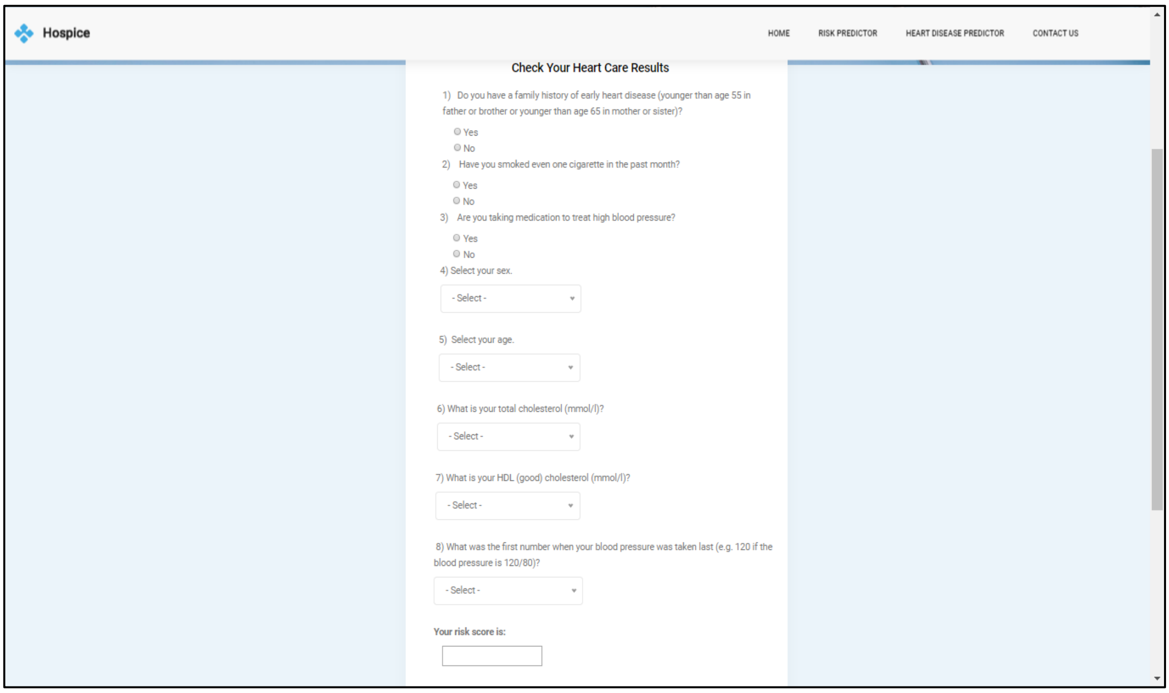
Eg. +91 XXXXXXXXXX

MESSAGE *

Write us a message

Fig.4.2 Output screen for Contact Us page

RISK PREDICTOR PAGE



Check Your Heart Care Results

1) Do you have a family history of early heart disease (younger than age 55 in father or brother or younger than age 65 in mother or sister)?

○ Yes
○ No

2) Have you smoked even one cigarette in the past month?

○ Yes
○ No

3) Are you taking medication to treat high blood pressure?

○ Yes
○ No

4) Select your sex.

- Select -

5) Select your age.

- Select -

6) What is your total cholesterol (mmol/l)?

- Select -

7) What is your HDL (good) cholesterol (mmol/l)?

- Select -

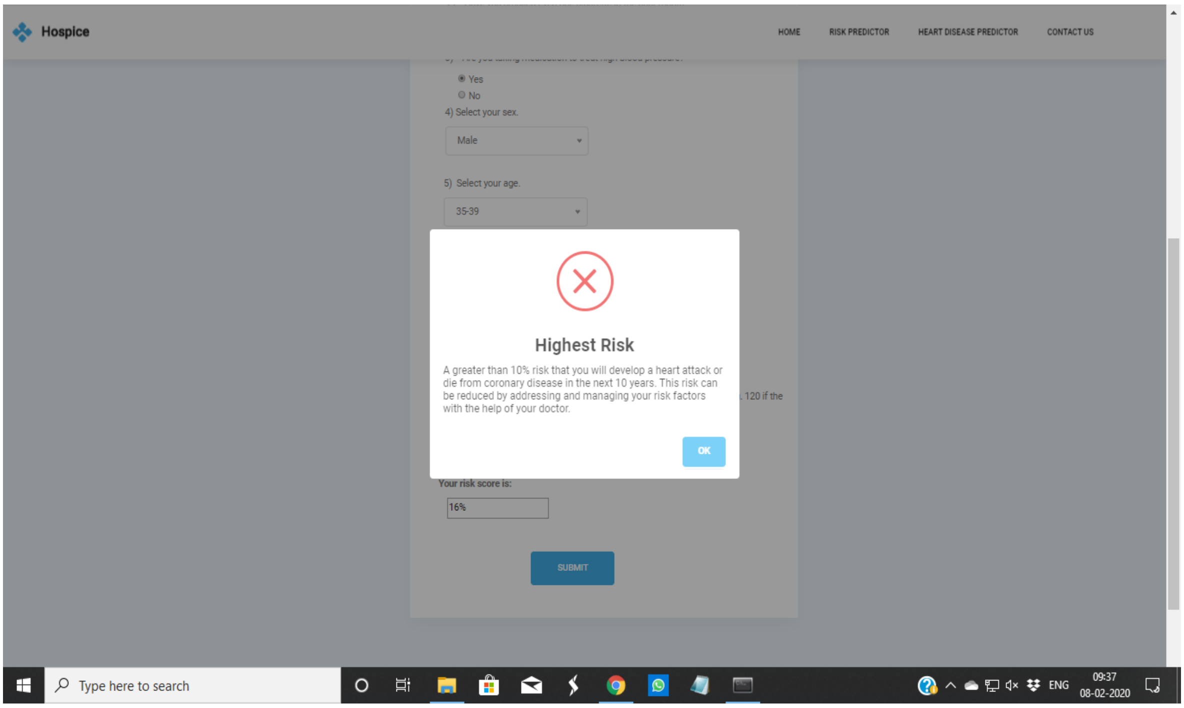
8) What was the first number when your blood pressure was taken last (e.g. 120 if the blood pressure is 120/80)?

- Select -

Your risk score is:

Fig.4.3 Output screen for Risk Predictor page

RISK PREDICTOR OUTPUT PAGE



3) Are you taking medication to treat high blood pressure?

● Yes
○ No

4) Select your sex.

Male ▾

5) Select your age.

35-39 ▾

. 120 if the

Highest Risk

A greater than 10% risk that you will develop a heart attack or die from coronary disease in the next 10 years. This risk can be reduced by addressing and managing your risk factors with the help of your doctor.

OK

Your risk score is:

16%

SUBMIT

Fig.4.4 Output screen for Risk Predictor Output page

HEART DISEASE PREDICTOR PAGE

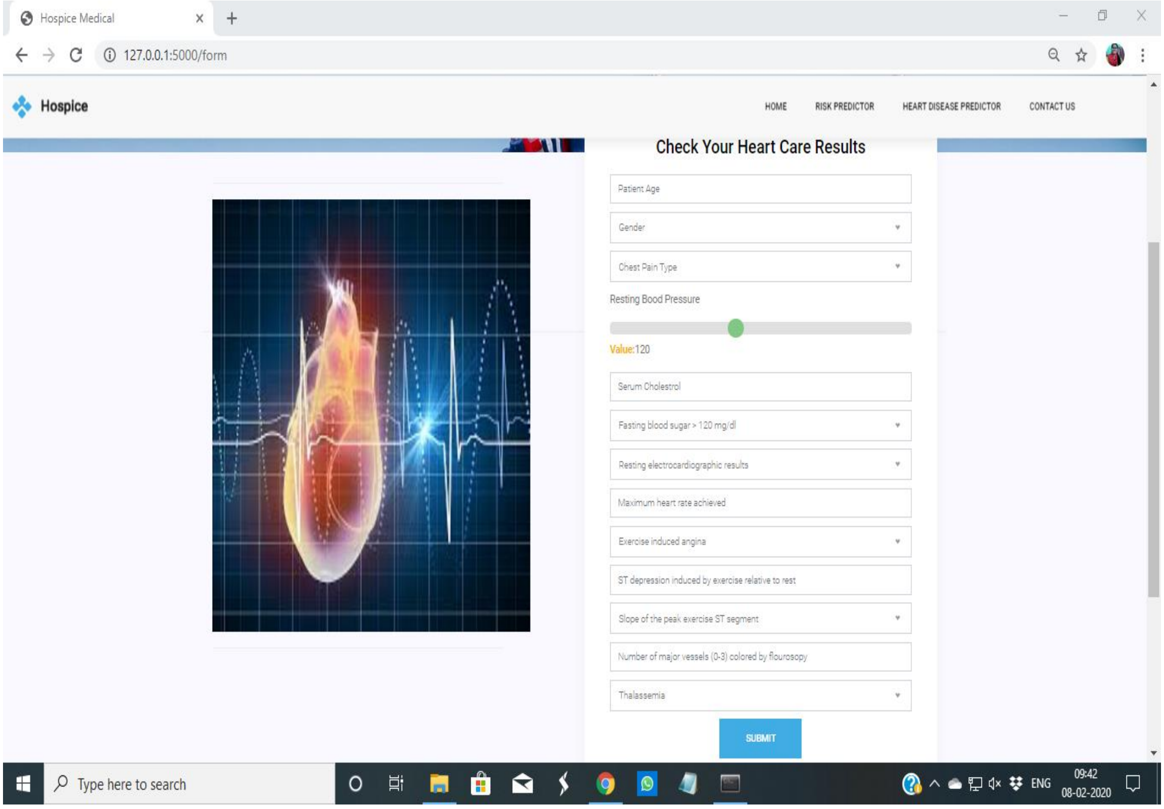
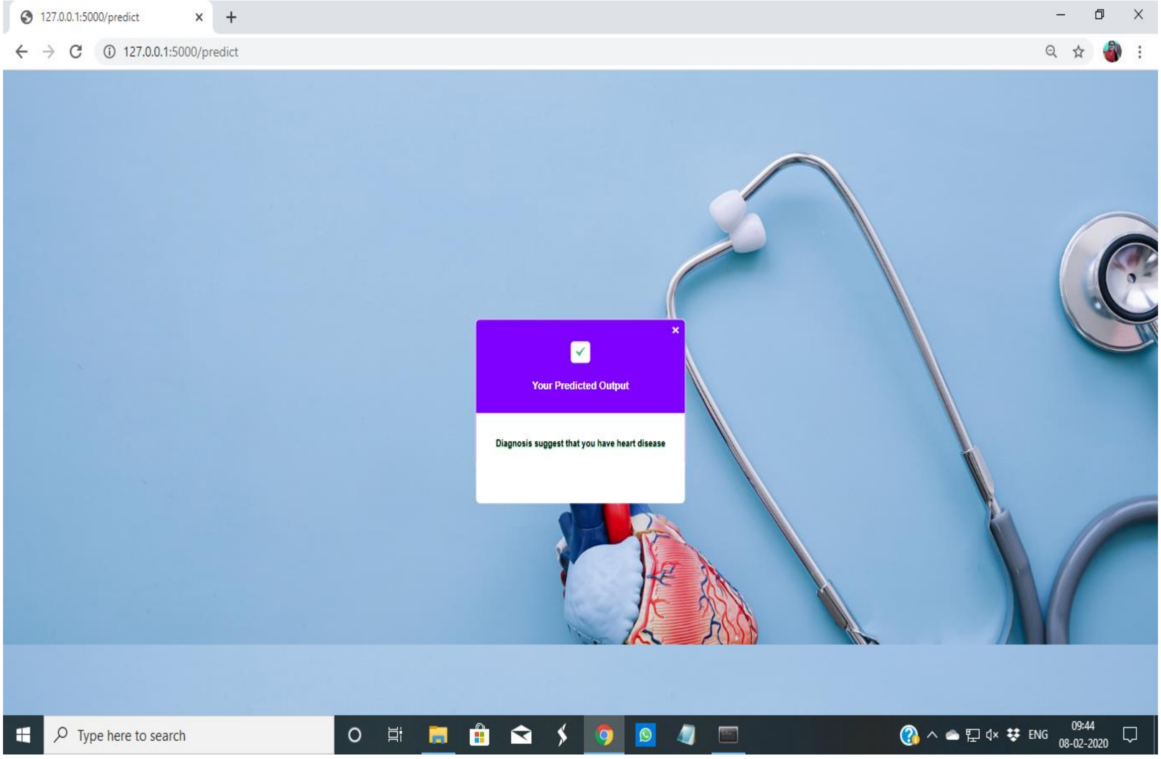

Hospice Medical
127.0.0.1:5000/form
Hospice
HOME
RISK PREDICTOR
HEART DISEASE PREDICTOR
CONTACT US
Check Your Heart Care Results
Patient Age
Gender
Chest Pain Type
Resting Bood Pressure
Value:120
Serum Cholestrol
Fasting blood sugar > 120 mg/dl
Resting electrocardiographic results
Maximum heart rate achieved
Exercise induced angina
ST depression induced by exercise relative to rest
Slope of the peak exercise ST segment
Number of major vessels (0-3) colored by flourosopy
Thalassemia
SUBMIT
Type here to search
ENG
09:42
08-02-2020

Fig.4.5 Output screen for Heart Disease Predictor page

HEART DISEASE PREDICTOR OUTPUT PAGE



Diagnosis suggest that you have heart disease

Fig.4.3 Output screen for Heart Disease Predictor Output page

7. CONCLUSION

We have used 5 algorithms like Decision Trees, Random Forests, Naive Bayes, KNN and Logistic Regression in-order to predict presence or absence of heart disease using SMOTE Technique. The accuracy varies for different algorithms. The accuracy for Decision tree algorithm is 75.71. The accuracy for Random Forest algorithm is 90. The accuracy for Naive Bayes algorithm is 90. The accuracy for KNN algorithm is 82.86. The highest accuracy is given when we have used Logistic Regression algorithm using SMOTE Technique which is nearly 94.29%.

8. FUTURE SCOPE

This project further can be enhanced by predicting disease and also suggest precautions to be taken by the person. This can also be developed as Android or IOS App.

9. REFERENCES

[1] www.who.int/cardiovascular_diseases/en/.

[2] http://food.ndtv.com/health/world-heart-day-20

[3] Jabbar MA, “ Prediction of heart disease using k nearest neighbour and particle swarm optimization”, in Biomed Res- India 2017 Volume 28 Issue 9, pp:4154-4158.

[4] Paul, Animesh Kumar & Chandra Shill, Pintu & Rabin, Md. Rafiqul Islam & Murase, Kazuyuki.,” Adaptive weighted fuzzy rule-based system for the risk level assessment of heart disease”, in Applied Intelligence.2017, 48. DOI:10.1007/s10489-017-1037-6.

[5] Jabbar MA, Deekshatulu BL,Priti C, “Prediction of risk score for heart disease using associative classification and hybrid feature Selection”, in IEEE ISDA 2012, pp:628-634.

[6] Oluwarotimi Williams Samuel , Grace Mojisola Asogbon , Arun Kumar Sangaiah , Fang Peng , Guanglin Li , “An Integrated Decision Support System Basedon ANN and Fuzzy_AHP for Heart Failure Risk Prediction” , in Expert Systems With Applications (2016), DOI: 10.1016/j.eswa.2016.10.020.

[7] Adeli A, Neshat M, “A fuzzy expert system for heart disease diagnosis”, In: Proceedings of the international multi-conference of engineers and computer scientists, vol I, 2010, pp: 1–6.

[8] Detrano R, Janosi A, Steinbrunn W, Pfisterer M, Schmid JJ, Sandhu S, Guppy KH, Lee S, Froelicher V , “International application of a new probability algorithm for the diagnosis of coronary artery disease”, in Am J Cardio, 64(5), 1989, pp:304–310, DOI:10.1016/0002-9149(89)90524-9 .

[9] Cheung N, “ Machine learning techniques for medical analysis.School of Information Technology and Electrical Engineering”, BSc Thesis, University of Queenland

[10] Das R, Turkoglu I, Sengur A, “Effective diagnosis of heart disease through neural networks ensembles”, in Expert Syst Appl, 2009, 36(4), pp:7675–7680, DOI:10.1016/j.eswa.2008.09.013

[11] Dangare A, “Data mining approach for prediction of heart disease using neural network”, in IJCET 2012; 3: 30-40.