**Heart Disease Prediction using Deep Learning**

A

MAJOR PROJECT REPORT

Submitted by

|  |  |  |
| --- | --- | --- |
| Manak | Manaswi | Pankaj Kumar |
| 06114802718 | 06214802718 | 07414802718 |

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

Ms. Garima Gupta

(Assistant Professor, CSE)



Department of Computer Science and Engineering

Maharaja Agrasen Institute of Technology,

PSP area, Sector – 22, Rohini, New Delhi – 110085

(Affiliated to Guru Gobind Singh Inderprastha University, New Delhi)

(JULY 2022)

**MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY**

Department of Computer Science and Engineering



**CERTIFICATE**

This is to certify that this MAJOR project report titled “Heart Disease Prediction using Deep Learning” is submitted by Manak (06114802718), Manaswi (06214802718) and Pankaj Kumar (07414802718) who carried out the project work under my supervision. I approve this Major Project.

|  |  |
| --- | --- |
| Prof. Namita Gupta  (HoD, CSE) | Ms Garima Gupta, Asisstant professor  (Project Guide) |

**ACKNOWLEDGEMENT**

It gives us immense pleasure to express our deepest sense of gratitude and sincere thanks to our respected guide, Ms. Garima Gupta, Assistant Professor, CSE, MAIT, Delhi, for their valuable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behaviour are sincerely acknowledged.

We also wish to express our indebtedness to our parents as well as our families whose blessings and support always helped us to face the challenges ahead.

|  |  |
| --- | --- |
| Place: Delhi  Date: | Manak  (06114802718) |
|  | Manaswi  (06214802718) |
|  | Pankaj Kumar  (07414802718) |

**ABSTRACT**

This document explores the possibility of the prediction whether a person is susceptible to various heart diseases like Coronary Artery Disease (CAD), Heart Arrhythmias, Heart Failure, Heart Valve Disease, Pericardial Disease, Cardiomyopathy (Heart Muscle Disease), Congenital Heart Disease and many more which has put a great threat to human beings given how our lives and schedules are evolving into more sedentary ones with the advent of technologies originally made to make our lives easier. A passive lifestyle puts not only our heart at risk, but also is a direct cause of more physical and mental illnesses and diseases like osteoporosis, lipid disorders diabetes, and obesity, and increase the risks of colon cancer, high blood pressure, depression and anxiety.

In this article, we have aimed to study the various different factors that may or may not be in direct correlation of heart diseases. These factors are as follows: Age, sex, chest pain type, resting blood pressure, cholesterol in mg/dl, fasting blood sugar, resting electrocardiography results, maximum heart rate achieved, exercise induced angina, ST depression induced by exercise, slope of the peak exercise ST segment, number of major vessels and maximum heart rate. We have also compared the correlation of these factors with the possibility of a heart related illness. These factors are elaborated in a more detailed way in this paper. And for the same, we have used multiple algorithms (logistic regression, naïve bayes, Support vector machine, KNN, decision tree, random forest and artificial neural network) and compare the results to find out the most accurate one. We are using dataset from kaggle.com.

Finally, we found out the results to be around 96% accurate so there is a 96% probability of the results being accurate when we feed custom data to the algorithm.

Keywords- angina, heart diseases, random forest, heart diseases prediction, classification

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TITLE** | **PAGE** |
| 1. | Title page | 1 |
| 2 | Certificate | 2 |
| 3 | Acknowledgement | 3 |
| 4 | Abstract | 4 |
| 5 | Table of contents | 5 |
| 6 | List of tables | 6 |
| 7 | List of figures | 7 |
| 8 | List of symbols and nomenclature | 8 |
| 9 | Chapter 1: Introduction | 9 |
| 10 | Chapter 2: Literature Survey |  |
| 11 | Chapter 3: Research approach |  |
| 12 | Chapter 4: Results |  |
| 13 | Chapter 5: Conclusion |  |
| 14 | References |  |
| 15 | Proof of research paper publishing |  |
| 16 |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TITLE** | **PAGE** |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **TITLE** | **PAGE** |
| 1 | Figure 1: Diagram of heart | 10 |
| 2 | Figure 2: Python Logo | 12 |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 | Figure 10: Confusion Matrix Sample | 22 |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

**LIST OF SYMBOLS AND NOMENCLATURE**

|  |  |
| --- | --- |
| **SYMBOL** | **MEANING** |
| CP | Chest pain |
| RESTBPS | Resting blood pressure |
| CHOL | Cholesterol level |
| FBS | Fasting blood sugar |
| RESTECG | Resting ECG |
| THALACH | Maximum heart rate |
| EXANG | Exercise induced angina |
| OLDPEAK | Depression in ST induced by exercise relative to rest |
| TP | True positives |
| TN | True negatives |
| FP | False positives |
| FN | False negatives |
| KNN | K-Nearest Neighbour |
| SVM | Support Vector Machine |
| CVD | CardioVascular Diseases |

**CHAPTER 1: INTRODUCTION**

This document explores the possibility of the prediction whether a person is susceptible to various heart diseases like Coronary Artery Disease (CAD), Heart Arrhythmias, Heart Failure, Heart Valve Disease, Pericardial Disease, Cardiomyopathy (Heart Muscle Disease), Congenital Heart Disease and many more which has put a great threat to human beings given how our lives and schedules are evolving into more sedentary ones with the advent of technologies originally made to make our lives easier. There could be other reasons behind a threatening heart disease as well including but not limiting to smoking, family history, high cholesterol, overwhelming stress and bad diets. A passive lifestyle puts not only our heart at risk, but also is a direct cause of more physical and mental illnesses and diseases like osteoporosis, lipid disorders diabetes, and obesity, and increase the risks of colon cancer, high blood pressure, depression and anxiety. As we all know, heart is a vital organ of our body. It has a very important task in our body, pumping blood to every part of our body. If it fails to function correctly, then the rest of the organs will stop working, and within a few minutes, the person will have a serious threat to their life.

Heart diseases and related illnesses are seen as one of the most prominent causes of death all around the world. According to the World Health Organisation, heart related diseases are responsible for taking 17.7 million lives every year, 31% of all global deaths. Situation in India too is not different, heart related diseases have become the leading cause of mortality. There has been a constant rise in the heart diseases related deaths in our country. The numbers rose around 34% from 1990 to 2016 [1]. According to the Global Burden of Disease study, CVD has a death rate of around 272 per 1,00,000 people which is significantly higher than the global average [2]. CVD directly increases the pressure on any nation’s health care and adversely impacts the productivity. As estimated by the World Health Organisation (WHO), CVD have costed India up to $237 billion, from 2005-2015 [3]. Thus, there is a grave need of a model that could predict, and in turn help prevent, the cardiovascular diseases. And that is exactly what we have endeavoured to do in this project.

In this research, we have aimed to study the various different factors that may or may not be in direct correlation of heart diseases. These factors are as follows: age, sex, chest pain type, resting blood pressure, cholesterol in mg/dl, fasting blood sugar, resting electrocardiography results, maximum heart rate achieved, exercise induced angina, ST depression induced by exercise, slope of the peak exercise ST segment, number of major vessels and maximum heart rate. We have also compared the correlation of these factors with the possibility of a heart related illness. These factors are elaborated in a more detailed way in this paper. And for the same, we have used the following algorithms

* Logistic regression
* Naïve bayes
* SVM
* KNN
* Decision tree
* Random forest
* Artificial neural network

In the end, we compare the outcomes and numbers from all these algorithm to find out the most suitable one. We are using dataset from kaggle.com.

Finally, we found out the results to be around 96% accurate so there is a 96% probability of the results being accurate when we feed custom data to the algorithm.

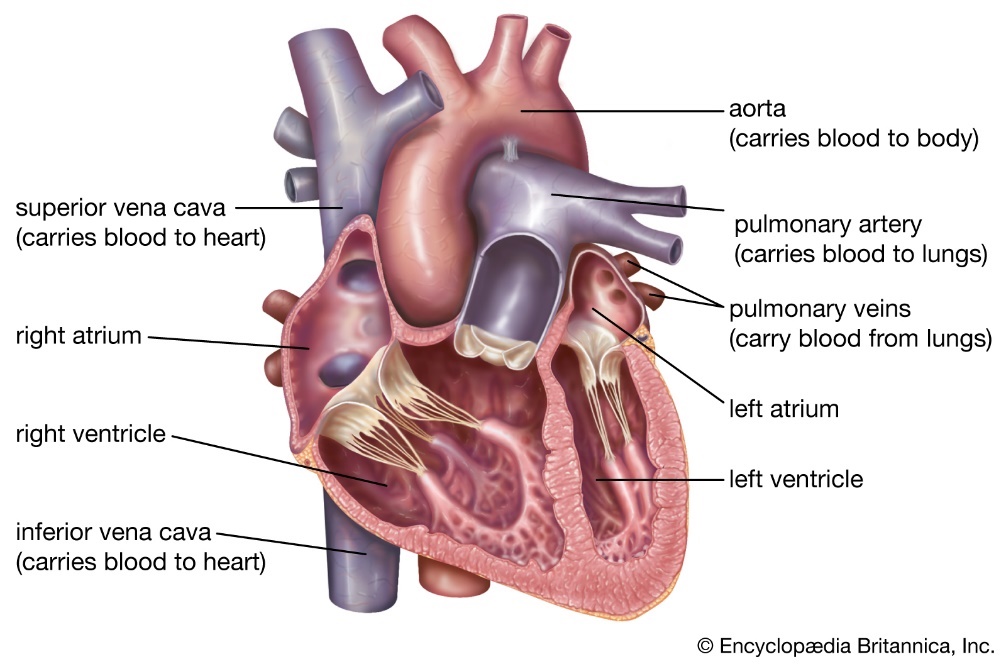


Figure 1: Diagram of heart

**CHAPTER 2: LITERATURE SURVEY**

There are many literature contributions under the same field, using a wide variety of technologies and methods to achieve better and efficient results. Xiao-Yan et al [4] used linear discriminant analysis (LDA) and principal component analysis (PCA) to extract the features and used 5 different classifier algorithms. They achieved the best accuracy of 98.6% with KNN when k =1. In the paper by Enriko et al [5], they achieve an accuracy of almost 82% and argue that the factors that we are including are sufficient enough for prediction. In [6], Boshra et al used K-Nearest Neighbors, J48 Decision tree, SMO and Naive Bayes and the 8 attributes with WEKA validation. The highest accuracy came out to be 83.7%. In a different study with a similar technique but different dataset, SVM delivered the best accuracy at 93.7 % and it seemed to be a robust design. [7].

In one article which used ANN with feature correlation analysis was studied on Sixth Korea National Health and Nutrition Examination dataset and the authors found that chronic renal failure and triglyceride were closely related to coronary heart disease, it showed the accuracy of 82.51%. [8]. In a study by Lavanya et al, J48, CART, Naïve Bayes were implemented using the WEKA tool to obtain an accuracy of almost 86% [9]. Chen et al used ANN observing multiple features and predicted a possibility of a cardiac arrest with an accuracy of 80% in article [10]. Haq et al in [11] talks about the usage of plethora of machine learning predictive models such as support vector machine (SVM), k-nearest neighbor, artificial neural network, decision tree, logistic regression, AdaBoos, Naive Bayes, fuzzy logic to get a best accuracy of 89% using LR with relief. In a study by Hidayet et al [12], 4 different feature selection methods and 12 classification algorithms has been used. This got the accuracy of around 85% using SVM and Naïve bayes.

If we compare ANN and neuro-fuzzy algorithms, as done by Abushariah et al [13], we discover that ANN has a better accuracy at 87% as compared to its counterpart which stands at around 76%. A three-phase model based on the ANN was proposed to achieve an 88.89% accuracy in a paper by Olaniyi et al [14]. When authors of another study proposed integrated decision based on ANN and Fuzzy AHP, the attained a 91.1% accuracy [15].

**CHAPTER 3: RESEARCH APPROACH**

3.1 Tools and Technologies used

The experiment has been performed using Intel (R) Core i7 CPU and 8 GB of memory.

3.1.1 Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. It supports functional and structured programming methods as well as OOP. It can be used as a scripting language or can be compiled to byte-code for building large applications. It provides very high-level dynamic data types and supports dynamic type checking. It supports automatic garbage collection. It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

|  |  |
| --- | --- |
| Python is:   * Easy-to-learn * Easy-to-read * Easy-to-maintain * A broad standard library * Interactive Mode * Portable * Extendable * GUI Programming * Scalable | Figure 2: Python Logo |

3.1.2 Jupyter Notebook

Project Jupyter is a project and community whose goal is to "develop [open-source software](https://en.wikipedia.org/wiki/Open-source_software), [open-standards](https://en.wikipedia.org/wiki/Open_standard), and services for [interactive computing](https://en.wikipedia.org/wiki/Interactive_computing) across dozens of programming languages". It was spun off from [IPython](https://en.wikipedia.org/wiki/IPython) in 2014 by [Fernando Pérez](https://en.wikipedia.org/wiki/Fernando_P%C3%A9rez_(software_developer)) and Brian Granger. Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)), and also a [homage](https://en.wikipedia.org/wiki/Homage_(arts)) to [Galileo](https://en.wikipedia.org/wiki/Galileo_Galilei)'s notebooks recording the discovery of the [moons of Jupiter](https://en.wikipedia.org/wiki/Moons_of_Jupiter). Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab. Jupyter is financially sponsored by NumFOCUS.

In 2014, Fernando Pérez announced a spin-off project from IPython called Project Jupyter. IPython continues to exist as a Python [shell](https://en.wikipedia.org/wiki/Command-line_interface) and a [kernel](https://en.wikipedia.org/wiki/Kernel_(operating_system)) for Jupyter, while the [notebook](https://en.wikipedia.org/wiki/Notebook_interface) and other [language-agnostic](https://en.wikipedia.org/wiki/Language-agnostic) parts of IPython moved under the Jupyter name. Jupyter is language agnostic and it supports execution environments (aka kernels) in several dozen languages among which are Julia, R, [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), and of course Python (via the IPython kernel). In 2015, [GitHub](https://en.wikipedia.org/wiki/GitHub) and the Jupyter Project announced native rendering of Jupyter notebooks file format (.ipynb files) on the GitHub platform.

3.1.3 Numpy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely. NumPy stands for Numerical Python. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also, it is optimized to work with latest CPU architectures. NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

3.1.4 Pandas

Pandas is an open-source library that is made mainly for working with relational or labeled data both easily and intuitively. It provides various data structures and operations for manipulating numerical data and time series. This library is built on top of the NumPy library. Pandas is fast and it has high performance & productivity for users.

Advantages

* Fast and efficient for manipulating and analyzing data.
* Data from different file objects can be loaded.
* Easy handling of missing data (represented as NaN) in floating point as well as non-floating-point data
* Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects
* Data set merging and joining.
* Flexible reshaping and pivoting of data sets
* Provides time-series functionality.
* Powerful group by functionality for performing split-apply-combine operations on data sets.

3.1.5 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Matplotlib comes with a wide variety of plots. Plots helps to understand trends, patterns, and to make correlations. They’re typically instruments for reasoning about quantitative information.

3.1.6 Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and colour palettes to make statistical plots more attractive. It is built on the top of [matplotlib](https://www.geeksforgeeks.org/python-introduction-matplotlib/) library and also closely integrated to the data structures from [pandas](https://www.geeksforgeeks.org/introduction-to-pandas-in-python/). Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Plots are basically used for visualizing the relationship between variables. Those variables can be either be completely numerical or a category like a group, class or division. Seaborn divides plot into the below categories – 

* Relational plots: This plot is used to understand the relation between two variables.
* [Categorical plots:](https://www.geeksforgeeks.org/seaborn-categorical-plots/)This plot deals with categorical variables and how they can be visualized.
* [Distribution plots:](https://www.geeksforgeeks.org/seaborn-distribution-plots/)This plot is used for examining univariate and bivariate distributions
* [Regression plots:](https://www.geeksforgeeks.org/seaborn-regression-plots/)The regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.
* [Matrix plots:](https://www.geeksforgeeks.org/ml-matrix-plots-in-seaborn/) A matrix plot is an array of scatterplots.
* Multi-plot grids: It is a useful approach is to draw multiple instances of the same plot on different subsets of the dataset.

3.1.7 Sklearn

Scikit-learn is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [programming language](https://en.wikipedia.org/wiki/Programming_language). It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support-vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [k-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy). Scikit-learn is a [NumFOCUS](https://en.wikipedia.org/w/index.php?title=NumFOCUS&action=edit&redlink=1) fiscally sponsored project.

The scikit-learn project started as scikits.learn, a [Google Summer of Code](https://en.wikipedia.org/wiki/Google_Summer_of_Code) project by French [data scientist](https://en.wikipedia.org/wiki/Data_scientist) [David Cournapeau](https://en.wikipedia.org/wiki/David_Cournapeau). Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to [SciPy](https://en.wikipedia.org/wiki/SciPy). The original [codebase](https://en.wikipedia.org/wiki/Codebase) was later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from the [French Institute for Research in Computer Science and Automation](https://en.wikipedia.org/wiki/French_Institute_for_Research_in_Computer_Science_and_Automation) in [Rocquencourt](https://en.wikipedia.org/wiki/Rocquencourt,_Yvelines), [France](https://en.wikipedia.org/wiki/France), took leadership of the project and made the first public release on February the 1st 2010. Of the various scikits, scikit-learn as well as [scikit-image](https://en.wikipedia.org/wiki/Scikit-image) were described as "well-maintained and popular" in November 2012. Scikit-learn is one of the most popular machine learning libraries on [GitHub](https://en.wikipedia.org/wiki/GitHub).

Scikit-learn is largely written in Python, and uses [NumPy](https://en.wikipedia.org/wiki/NumPy) extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in [Cython](https://en.wikipedia.org/wiki/Cython) to improve performance. Support vector machines are implemented by a Cython wrapper around [LIBSVM](https://en.wikipedia.org/wiki/LIBSVM); logistic regression and linear support vector machines by a similar wrapper around [LIBLINEAR](https://en.wikipedia.org/wiki/LIBLINEAR). In such cases, extending these methods with Python may not be possible.

Scikit-learn integrates well with many other Python libraries, such as [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib) and [plotly](https://en.wikipedia.org/wiki/Plotly) for plotting, [NumPy](https://en.wikipedia.org/wiki/NumPy) for array vectorization, [Pandas](https://en.wikipedia.org/wiki/Pandas_(software)) dataframes, [SciPy](https://en.wikipedia.org/wiki/SciPy), and many more.

3.1.8 Keras

Keras is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software](https://en.wikipedia.org/wiki/AI_software) library that provides a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) interface for [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network). Keras acts as an interface for the [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow) library.

Up until version 2.3, Keras supported multiple backends, including [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow), [Microsoft Cognitive Toolkit](https://en.wikipedia.org/wiki/Microsoft_Cognitive_Toolkit), [Theano](https://en.wikipedia.org/wiki/Theano_(software)), and [PlaidML](https://en.wikipedia.org/wiki/PlaidML). As of version 2.4, only [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow) is supported. Designed to enable fast experimentation with [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning), it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is [François Chollet](https://en.wikipedia.org/wiki/Fran%C3%A7ois_Chollet), a [Google](https://en.wikipedia.org/wiki/Google) engineer. Chollet is also the author of the XCeption deep neural network model.

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, [objectives](https://en.wikipedia.org/wiki/Objective_function), [activation functions](https://en.wikipedia.org/wiki/Activation_function), [optimizers](https://en.wikipedia.org/wiki/Mathematical_optimization), and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on [GitHub](https://en.wikipedia.org/wiki/GitHub), and community support forums include the GitHub issues page, and a [Slack](https://en.wikipedia.org/wiki/Slack_(software)) channel.

In addition to standard neural networks, Keras has support for [convolutional](https://en.wikipedia.org/wiki/Convolutional_neural_networks) and [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks). It supports other common utility layers like [dropout](https://en.wikipedia.org/wiki/Dropout_(neural_networks)), [batch normalization](https://en.wikipedia.org/wiki/Batch_normalization), and [pooling](https://en.wikipedia.org/wiki/Pooling_(neural_networks)).

Keras allows users to productize deep models on smartphones ([iOS](https://en.wikipedia.org/wiki/IOS) and [Android](https://en.wikipedia.org/wiki/Android_(operating_system))), on the web, or on the [Java Virtual Machine](https://en.wikipedia.org/wiki/Java_Virtual_Machine). It also allows use of distributed training of deep-learning models on clusters of [Graphics processing units (GPU)](https://en.wikipedia.org/wiki/Graphics_processing_unit) and [tensor processing units (TPU)](https://en.wikipedia.org/wiki/Tensor_processing_unit).

3.2 The attributes

1. age (#)  
2. sex : 1= Male, 0= Female (Binary)  
3. (cp) chest pain type (4 values -Ordinal):

Value 1: typical angina ,

Value 2: atypical angina,

Value 3: non-anginal pain ,

Value 4: asymptomatic

4. (trestbps) resting blood pressure (#)  
5. (chol) serum cholesterol in mg/dl (#)  
6. (fbs)fasting blood sugar > 120 mg/dl(Binary)(1 = true; 0 = false)  
7. (restecg) resting electrocardiography results(values 0,1,2)  
8. (thalach) maximum heart rate achieved (#)  
9. (exang) exercise induced angina (binary) (1 = yes; 0 = no)  
10. (oldpeak) = ST depression induced by exercise relative to rest (#)  
11. (slope) of the peak exercise ST segment (Ordinal)

Value 1: up sloping,

Value 2: flat.

Value down sloping.  
12. (ca) number of major vessels (0–3, Ordinal) colored by fluoroscopy  
13. (thal) maximum heart rate achieved — (Ordinal):

3 = normal;

6 = fixed defect;

7 = reversible defect

Note: Our data has 3 types of data:

Continuous (#): which is quantitative data that can be measured

Ordinal Data: Categorical data that has a order to it (0,1,2,3, etc)

Binary Data: data whose unit can take on only two possible states (0 &1)

3.3 Dataset

The dataset used in this project was gathered from kaggle.com. It is a combination of multiple datasets combined to form one to make a bigger dataset so as to give more training points to the algorithm. In this section, we discuss about the dataset used and some statistics of our dataset.

3.3.1 Shape and sample from the dataset

In this we talk about the size of the dataset used. We have attached screenshots from the Jupyter notebook, depicting the various statistics and information about the dataset. This information is very important and absolutely crucial since we should have and understanding of our dataset before we put it to some use. Datasets are fundamental to foster the development of several computational fields, giving scope, robustness, and confidence to results. And not only that, but a Good dataset allows for efficient analysis, limits errors and inaccuracies that can occur to data during processing, and makes all processed data more accessible to users. It's also gotten easier with new tools that enable any user to cleanse and qualify data on their own.

3.3.2 Correlation of the attributes

Now, we discuss about the correlation of the different attributes of our dataset and we use seaborn library to depict the heatmap which displays the correlations between attributes using the different shades of colours. Here, the darker shade implies that the two attributes are more related to each other as compared to the other attributes. All the values in this map are between 0 and 1. And diagonally, we see that correlation is ‘1’ because an attribute has the highest relation to itself, hence we see one and the darkest shade in that box.

3.3.3 Exploratory data analysis

In the final subsection of this section, we study our datasets. We will talk about how they vary and how they look plotted against our “target” variable. We will keep the target value separated from our rest of dataset so we can study different trends in the numbers and make a general assumption about how different values can affect our target variable.

1. Target

This variable shows whether the person in our dataset has a heart disease or not. This is the most crucial attribute as it helps in training and testing our algorithm.int the below countplot made using matplotlib, we see the distribution of people suffering from heart diseases and those who do not. We notice how both number of people are almost equal. This fact makes sure there is no bias and our algorithm gives out the as accurate results as possible.

1. Age

Age is the most important risk factor in developing cardiovascular or heart diseases, with approximately a tripling of risk with each decade of life. Coronary fatty streaks can begin to form in adolescence. It is estimated that 82 percent of people who die of coronary heart disease are 65 and older. Simultaneously, the risk of stroke doubles every decade after age 55.

1. Sex

In this plot, we observe how more men in our dataset suffers from a heart disease as compared to women. It is also an accepted fact that men are more likely to get a heart disease as compared to women.

1. Chest pain

Angina is chest pain or discomfort caused when your heart muscle doesn’t get enough oxygen-rich blood. It may feel like pressure or squeezing in your chest. The discomfort also can occur in your shoulders, arms, neck, jaw, or back. Angina pain may even feel like indigestion. In our dataset, we have divided this category into four parts. (0) asymptomatic, (2) atypical angina, (3) non-anginal pain and (4) typical angina

1. Fasting blood sugar

Not producing enough of a hormone secreted by your pancreas (insulin) or not responding to insulin properly causes your body’s blood sugar levels to rise, increasing your risk of a heart attack. In our dataset, if blood sugar level is above 120ml/dl, it is represented by 1, otherwise by 0.

1. Resting ECG

For people at low risk of cardiovascular disease, the USPSTF concludes with moderate certainty that the potential harms of screening with resting or exercise ECG equal or exceed the potential benefits. For people at intermediate to high risk, current evidence is insufficient to assess the balance of benefits and harms of screening.

1. Exercise induced Angina

The pain or discomfort associated with angina usually feels tight, gripping or squeezing, and can vary from mild to severe. Angina is usually felt in the center of your chest but may spread to either or both of your shoulders, or your back, neck, jaw or arm. It can even be felt in your hands. o Types of Angina a. Stable Angina / Angina Pectoris b. Unstable Angina c. Variant (Prinzmetal) Angina d. Microvascular Angina.

1. Slope of the peak exercise ST segment

A treadmill ECG stress test is considered abnormal when there is a horizontal or down-sloping ST-segment depression ≥ 1 mm at 60–80 ms after the J point. Exercise ECGs with up-sloping ST-segment depressions are typically reported as an ‘equivocal’ test. In general, the occurrence of horizontal or down-sloping ST-segment depression at a lower workload (calculated in METs) or heart rate indicates a worse prognosis and higher likelihood of multi-vessel disease. The duration of ST-segment depression is also important, as prolonged recovery after peak stress is consistent with a positive treadmill ECG stress test. Another finding that is highly indicative of significant CAD is the occurrence of ST-segment elevation > 1 mm (often suggesting transmural ischemia); these patients are frequently referred urgently for coronary angiography.

1. Number of major blood vessels

A high level of low-density lipoprotein (LDL) cholesterol (the “bad” cholesterol) is most likely to narrow arteries. A high level of triglycerides, a type of blood fat related to your diet, also ups your risk of a heart attack. However, a high level of high-density lipoprotein (HDL) cholesterol (the “good” cholesterol) lowers your risk of a heart attack.

1. Maximum heart rate achieved

The increase in cardiovascular risk, associated with the acceleration of heart rate, was comparable to the increase in risk observed with high blood pressure. It has been shown that an increase in heart rate by 10 beats per minute was associated with an increase in the risk of cardiac death by at least 20%, and this increase in the risk is similar to the one observed with an increase in systolic blood pressure by 10 mm Hg.

3.4 Classification

Now, we will use the following classification algorithms and feed them our dataset to train the algorithms and study the outcomes of each algorithm.

1. Logistic regression
2. Naïve Bayes
3. SVM
4. K-Nearest Neighbours
5. Neural network
6. Decision Tree
7. Random Forest

But before that, we drop the “target” attribute from our data and rename the target as “Y” and the rest of data as “X”. We do that to help the training and testing process of the algorithms easier. After that, we split the data into training and testing sets, which finally divides the data into four parts, namely, X\_train, X\_test, Y\_train and Y\_test. This is done with the help of sklearn library method called “train\_test\_split”. Then we will use multiple parameters to judge a model. These parameters are listed as below:

1. Confusion matrix

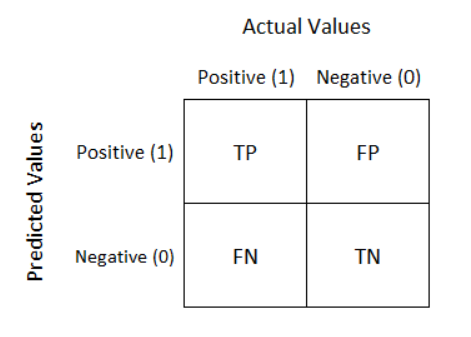


Figure 10: Confusion matrix Sample

Classification is the process of categorizing a given set of data into classes.  
In Machine Learning (ML), you frame the problem, collect and clean the data, add some necessary feature variables (if any), train the model, measure its performance, improve it by using some cost function, and then it is ready to deploy.   
But how do we measure its performance? Is there any particular feature to look at?  
A trivial and broad answer would be to compare the actual values to the predicted values. But that does not solve the issue.

1. Accuracy

Accuracy is the ratio of the True predicted values to the Total predicted values.

1. Precision

The precision for class 1 is, out of all predicted class values like 1, how many actually belong to class 1.

1. Recall

Recall for class 1 is, out of all the values that actually belong to class 1, how much is predicted as class 1.

1. F1

To have a combined effect of precision and recall, we use the F1 score. The F1 score is the harmonic mean of precision and recall.

**CHAPTER 4: RESULTS**

In this chapter, we are going to run the classification algorithms and find out the numbers for each one of them to compare the best algorithm. We will compare the confusion matrix and other scores of every algorithm.

1. 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.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

1. Naïve Bayes

Naïve Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with a strong assumption that all the predictors are independent to each other i.e. the presence of a feature in a class is independent to the presence of any other feature in the same class. This is naïve assumption that is why these methods are called Naïve Bayes methods.

Bayes theorem states the following relationship in order to find the posterior probability of class i.e. the probability of a label and some observed features, P(Y⏐features)P(Y⏐features).

P(Y⏐features)=(P⟮Y⟯P(features⏐Y)P(features))P(Y⏐features)=(P⟮Y⟯P(features⏐Y)P(features))

Here, P(Y⏐features)P(Y⏐features) is the posterior probability of class.

P(Y)P(Y) is the prior probability of class.

P(features⏐Y)P(features⏐Y) is the likelihood which is the probability of predictor given class.

P(features)P(features) is the prior probability of predictor.

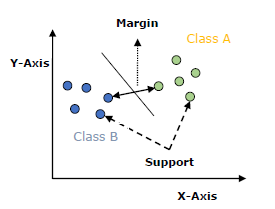
The Scikit-learn provides different naïve Bayes classifiers models namely Gaussian, Multinomial, Complement and Bernoulli. All of them differ mainly by the assumption they make regarding the distribution of 𝑷P(features⏐Y)P(features⏐Y) i.e. the probability of predictor given class.

1. SVM

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

Working of SVM

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).



The followings are important concepts in SVM −

Support Vectors − Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.

Hyperplane − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

1. K-Nearest Neighbours

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well −

Lazy learning algorithm − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.

Non-parametric learning algorithm − KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.

Working of KNN Algorithm

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following −

3.1 − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.

3.2 − Now, based on the distance value, sort them in ascending order.

3.3 − Next, it will choose the top K rows from the sorted array.

3.4 − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 – End

1. Neural network

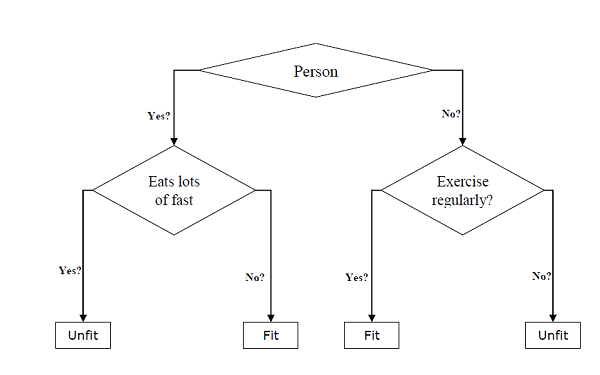
Neural networks are parallel computing devices, which are basically an attempt to make a computer model of the brain. The main objective is to develop a system to perform various computational tasks faster than the traditional systems. This tutorial covers the basic concept and terminologies involved in Artificial Neural Network. Sections of this tutorial also explain the architecture as well as the training algorithm of various networks used in ANN.

ANN is an advanced topic, hence the reader must have basic knowledge of Algorithms, Programming, and Mathematics.

1. Decision Tree

In general, Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decisions tress are the most powerful algorithms that falls under the category of supervised algorithms.

They can be used for both classification and regression tasks. The two main entities of a tree are decision nodes, where the data is split and leaves, where we got outcome. The example of a binary tree for predicting whether a person is fit or unfit providing various information like age, eating habits and exercise habits, is given below −



In the above decision tree, the question are decision nodes and final outcomes are leaves. We have the following two types of decision trees −

Classification decision trees − In this kind of decision trees, the decision variable is categorical. The above decision tree is an example of classification decision tree.

Regression decision trees − In this kind of decision trees, the decision variable is continuous.

1. Random Forest

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

We can understand the working of Random Forest algorithm with the help of following steps −

Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

The following diagram will illustrate its working −



**CHAPTER 5: CONCLUSIONS**

**REFERENCES**

[1] [Heart disease deaths rise in India by 34% in 26 years | India News - Times of India (indiatimes.com)](https://timesofindia.indiatimes.com/india/heart-disease-deaths-rise-in-india-by-34-in-15-years/articleshow/64924601.cms#:~:text=1%20The%20death%20rate%20due%20to%20cardiovascular%20diseases,of%20all%20deaths%20in%20India%20and%20the%20US)

[2] [Cardiovascular disease in India: A 360 degree overview (nih.gov)](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6994761/)

[3] [Dorairaj](https://www.ahajournals.org/doi/10.1161/circulationaha.114.008729) P., [Panniyammakal J](https://www.ahajournals.org/doi/10.1161/circulationaha.114.008729).and [Ambuj R](https://www.ahajournals.org/doi/10.1161/circulationaha.114.008729)., Cardiovascular Diseases in India, 19 Apr 2016 <https://doi.org/10.1161/CIRCULATIONAHA.114.008729>Circulation. 2016;133:1605–1620

[4] Mostafa A, Xiao-Yan G, Amin Ali, Hassan S, Eman M., (2021), Improving the Accuracy for Analyzing Heart Diseases Prediction Based on the Ensemble Method, 6663455, 2021/02/10, 1076-2787, https://doi.org/10.1155/2021/6663455, Hindawi

[5] Enriko I. K. A., Muhammad, Suryanegara. and Dadang, Gunawan. (2016). Heart Disease Prediction System using k-Nearest Neighbor Algorithm with Simplified Patient's Health Parameters. Journal of Telecommunication, Electronic and Computer Engineering, 8(12), 59-65.

[6] Boshra, Bahrami., Mirsaeid, Hosseini, Shirvani. (2015). Prediction and Diagnosis of Heart Disease by Data Mining Techniques. Journal of Multidisciplinary Engineering Science and Technology, 2(2), 164-168.

[7] Janardhanan, P., L, Heena. and Sabika, F. (2015). Effectiveness of Support Vector Machines in Medical Data Mining. Journal of Communications Software and Systems, 11(1), 25-30.

[8] Karayılan, T. and Kılıç, Ö. (2017). Prediction of heart disease using neural network. International Conference on Computer Science and Engineering (UBMK), Antalya, 719-723. doi: 10.1109/UBMK.2017.8093512.

[9] Lavanya, M., Gomathi, P, M. (2016). Prediction of Heart Disease using Classification Algorithms. International Journal of Advanced Research in Computer Engineering & Technology, 5(7), 2173- 2175.

[10] A. H. Chen, S.-Y. Huang, P.-S. Hong, C.-H. Cheng, and E.-J. Lin, “Hdps: heart disease prediction system,” In 2011 computing in Cardiology, vol. 557–560, 2011.View at: [Google Scholar](https://scholar.google.com/scholar_lookup?title=Hdps:%20heart%20disease%20prediction%20system&author=A.%20H.%20Chen&author=S.-Y.%20Huang&author=P.-S.%20Hong&author=C.-H.%20Cheng&author=&author=E.-J.%20Lin&publication_year=2011)

[11] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, and R. Sun, “A hybrid intelligent system framework for the prediction of heartdisease using machine learning algorithms,” Mobile Information Systems, vol. 2018, Article ID 3860146, 21 pages, 2018.View at: [Publisher Site](https://doi.org/10.1155/2018/3860146) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=A%20hybrid%20intelligent%20system%20framework%20for%20the%20prediction%20of%20heartdisease%20using%20machine%20learning%20algorithms&author=A.%20U.%20Haq&author=J.%20P.%20Li&author=M.%20H.%20Memon&author=S.%20Nazir&author=&author=R.%20Sun&publication_year=2018)

[12] Hidayet, Takci. (2018). Improvement of Heart Attack Prediction by the Feature Selection Methods. Turk J Elec Eng & Comp Sci, 26, 1-10

[13] M. A. Abushariah, A. A. Alqudah, O. Y. Adwan et al., “Automatic heart disease diagnosis system based onartificial neural network (ann) and adaptive neuro-fuzzy inference systems (anfis) approaches,” Journal of Software Engineering and Applications, vol. 7, p. 1055, 2014.View at: [Publisher Site](https://doi.org/10.4236/jsea.2014.712093) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=Automatic%20heart%20disease%20diagnosis%20system%20based%20onartificial%20neural%20network%20(ann)%20and%20adaptive%20neuro-fuzzy%20inference%20systems%20(anfis)%20approaches&author=M.%20A.%20Abushariah&author=A.%20A.%20Alqudah&author=O.%20Y.%20Adwan%20et%20al.&publication_year=2014)

[14] E. O. Olaniyi and O. K. Oyedotun, “Heart diseases diagnosis using neural networks arbitration,” International Journal of Intelligent Systems and Applications, vol. 7, no. 12, pp. 75–82, 2015.View at: [Publisher Site](https://doi.org/10.5815/ijisa.2015.12.08) | [Google Scholar](https://scholar.google.com/scholar_lookup?title=Heart%20diseases%20diagnosis%20using%20neural%20networks%20arbitration&author=E.%20O.%20Olaniyi%20&author=O.%20K.%20Oyedotun&publication_year=2015)

[15] O. W. Samuel, G. M. Asogbon, A. K. Sangaiah, P. Fang, and G. Li, “An integrated decision support system based on ANN and Fuzzy\_AHP for heart failure risk prediction,” Expert Systems with Applications, vol. 68, pp. 163–172, 2017.

[16] wikipedia.org

[17] geeksforgeeks.org

[18] w3schools.com