Prediction of Cardiovascular Disease using Machine and Deep Learning

Final Year Project Report

Presented

By

Aieza Noor

CIIT/FA20-BSM-002/ISB

In Partial Fulfillment
Of the Requirement for the Degree of
Bachelor of Science in Mathematics

DEPARTMENT OF MATHEMATICS

COMSATS UNIVERSITY ISLAMABAD

May 2024

Prediction of Cardiovascular Disease using Machine and Deep Learning

Final Year Project Report

By

Aieza Noor

CIIT/FA20-BSM-002/ISB

In Partial Fulfillment

Of the Requirement for the Degree of
Bachelor of Science in Mathematics

DEPARTMENT OF MATHEMATICS

COMSATS UNIVERSITY ISLAMABAD

May 2024

Declaration

I, Aieza Noor, registration number CIIT/FA20-BSM-002/ISB *hereby declare that I have developed this project and the accompanied report entirely on the basis of my personal efforts made under the sincere guidance of my supervisor. No portion of the work presented in this report has been submitted in the support of any other degree or qualification of this or any other University or Institute of learning, if found I shall stand responsible.*

Signature: _____

Aieza Noor

May 2024

COMSATS UNIVERSITY ISLAMABAD

May 2024

Prediction of Cardiovascular Disease using Machine and Deep Learning

An Undergraduate Final Year Project Report submitted to the

Department of MATHEMATICS

As a Partial Fulfillment for the award of Degree
Bachelor of Science in Mathematics

By

Name	Registration Number
Aieza Noor	CIIT/FA20-BSM-002/ISB

Supervised by

Dr. Iftikhar Ahmed

Assistant Professor

Department Of Mathematics

COMSATS UNIVERSITY ISLAMABAD

May 2024

Final Approval

This Project Titled

Prediction of Cardiovascular Disease using Machine and Deep Learning

Submitted for the Degree of
Bachelor of Science in Mathematics

By

Name	**Registration Number**
Aieza Noor	CIIT/FA20-BSM-002/ISB

has been approved for

COMSATS UNIVERSITY ISLAMABAD

*Supervisor*_____

Dr. Iftikhar Ahmed

Assistant Professor

*External Examiner*_____

Prof. Dr. Salman Ahmad

*Head of Department*_____

Prof. Dr. Shamsul Islam

Dedication

This work is dedicated to

My beloved Parents, Teachers

And all those who care for me

Acknowledgement

I bow my head with the deepest gratitude to Almighty Allah, who enabled me to complete this piece of work. He is the most powerful, compassionate, kind and merciful. I offer my humblest words of thanks to the Holy Prophet Muhammad (peace be upon him) who is forever a torch of guidance for humanity.

My sincere gratitude to my Supervisor Dr. Iftikhar Ahmed for his friendly approach, readiness, and willingness to advise on the research, commitment to guide, and support greatly improved this thesis work. It was a pleasure working with him and a wonderful learning experience. Last but not the least, I would like to thank my uncle Dr. Tanvir Akbar Kiani and my parents for their continuous support, understanding and assistance whenever I needed them throughout my BS studies and research work. I believe that without their motivation, it is not possible to succeed throughout my life. I am always grateful to them for their encouragement and support. Also, thanks to my friends Aqdas Bibi and Aqsa Sarfraz for always standing and guiding me in my academics and life in general and thanks to my classmates and friends for their support and love.

Aieza Noor

Abstract

Prediction of Cardiovascular Disease using Machine and Deep Learning

Cardiovascular diseases (CVD) are principal factors of deaths worldwide and its early prediction is important. During recent years, prediction of CVDs grabbed much attention which led to different researches. Predicting and preventing cardiovascular diseases can result in considerable cost saving for individuals and healthcare systems. Deep learning (DL) has played a vital role in predicting and diagnosing different diseases. In this project, we used different machine learning techniques and deep learning models for the accurate prediction of cardiovascular diseases. Results are shown individually to provide comparison. Publicly available dataset is taken from kaggle. Among all the algorithms, Gated Recurrent Unit (GRU) achieved an accuracy of 73.92%.

Contents

Chapter 1

Introduction

1.1 Introduction

There is significant increase in rate of deaths worldwide due to Cardiovascular diseases. According to a survey by World Health Organization (WHO), about 17.9 million individuals die annually due to cardiovascular disease. Cardiovascular disease (CVD) is a general term for conditions affecting the heart or blood vessels. It's usually associated with a build-up of fatty deposits inside the arteries (atherosclerosis) and an increased risk of blood clots. Identifying CVD in its early stage can be significant in preventing and treating the disease effectively. Approximately half of all patients diagnosed with Heart Disease die within just 1-2 years, while merely 3% of the total budget for health care is deployed on treating heart disease. To predict heart disease multiple tests are required. Lack of expertise of medical staff may results in false predictions [1]. Several cardiovascular diseases (CVD) pose significant threats to human life, often arising from detectable risk factors such as tobacco use, poor dietary habits, physical dormancy, and excessive alcohol consumption across various setting. Human beings who are having CVD are at high risk of cardiac [2].

This study aims to identify the significant features and Machine learning techniques and deep learning models to predict cardiovascular disease.

The rest of the document is structured as follows: Chapter-1 describes the existing literature work. Chapter-2 gives an overview of the proposed methodology. Chapter-3 explains the background of the machine learning algorithms and deep

learning models and their functionalities. Finally, experimental results are discussed in Chapter-4.

1.2 Problem Statement

Despite advancements in medical science, accurately predicting cardiovascular diseases remains a challenge due to various factors including environmental and lifestyle factors. This study aims to utilize a healthcare dataset, complex computational techniques to boost patient well-being foster advancement in cardiovascular healthcare.

1.3 Literature Review

1.3.1 Key findings of Research papers

Source	Key Findings
Sajja, T.K. and Kalluri, H.K., 2020. A Deep Learning Method for Prediction of Cardiovascular Disease Using Convolutional Neural Network. Rev. d'Intelligence Artif., 34(5), pp.601-606 [2].	• Since machine learning algorithms are not enough to detect diseases, a deep learning approach is used to predict the disease caused by heart block • Machine learning techniques: Logistic regression, KNN, Naïve Bayes, Support Vector Machine • Proposed CNN to predict cardiovascular disease at an early stage • Compared traditional approaches and proposed CNN model • Got an accuracy of 94%
Li, Y., He, Z., Wang, H., Li, B., Li, F., Gao, Y. and Ye, X., 2020. CraftNet: a deep learning ensemble to diagnose cardiovascular diseases. Biomedical Signal Processing and Control, 62, p.102091 [3].	• Proposed a deep neural network named CraftNet • Study is mainly focused on handcraft and deep features for analysis of cardiovascular diseases from electro cardio graph(ECG) • The proposed model accurately recognizes the handcraft features • The proposed model has stronger classification ability and is less affected by data imbalance • Average sensitivity accuracy increased from 86.82% to 89.25%

Source	Key Findings
Johri, A.M., Singh, K.V., Mantella, L.E., Saba, L., Sharma, A., Laird, J.R., Utkarsh, K., Singh, I.M., Gupta, S., Kalra, M.S. and Suri, J.S., 2022. Deep learning artificial intelligence framework for multiclass coronary artery disease prediction using combination of conventional risk factors, carotid ultrasound, and intraplaque neovascularization. Computers in Biology and Medicine, 150, p.106018 [4].	• Proposed $AtheroEdge - MCDL_{AI}$ windows-based platform using multiclass Deep Learning (DL) system • Data of 500 patients is collected • Total 39 covariates were used including office-based, laboratory-based, and carotid ultrasound image phenotypes • Deep learning models: Recurrent Neural Network (RNN),LSTM • Used SMOTE as over-sampling technique for the handling imbalanced data • Computed performance using 10-fold validation technique • Got an accuracy of 95.00% and 95.34% for RNN and LSTM
Pal, M., Parija, S., Panda, G., Dhama, K. and Mohapatra, R.K., 2022. Risk prediction of cardiovascular disease using machine learning classifiers. Open Medicine, 17(1), pp.1100-1113 [5].	• Publicly available university of California Irvine repository data is taken for the experiment to detect CVD • Machine learning techniques: Multi-layer perceptron (MLP) and K-nearest neighbors (KNN) • Results demonstrate an accuracy of 82.47

Source	Key Findings
	• Proposed MLP model is recommended for automatic CVD detection
Himi, S.T., Monalisa, N.T., Whaiduzzaman, M.D., Barros, A. and Uddin, M.S., 2023. MedAi: A smartwatch-based application framework for the prediction of common diseases using machine learning. IEEE Access, 11, pp.12342-12359 [6].	• A smartwatch-based prediction system named "MedAi" is presented in this paper to predict many diseases using machine learning algorithms • This smartwatch prediction system uses eight machine learning algorithms to predict twelve different kinds of diseases displayed through an android mobile application • This system is comprised of three main modules: a smartwatch equipped with sensors to collect bodily statistics, a data analyzing machine learning algorithm to make a prediction and a mobile application to display the result • Patients' bodily statistics dataset was collected from a local hospital • Several experimentations on collected dataset shows that Random Forest outperforms other machine learning algorithms with an accuracy of 99.4%

Source	Key Findings
Al Reshan, M.S., Amin, S., Zeb, M.A., Sulaiman, A., Alshahrani, H. and Shaikh, A., 2023. A Robust Heart Disease Prediction System Using Hybrid Deep Neural Networks. IEEE Access [7].	• Three deep models are use: Artificial Neural Networks, Convolutional Neural Networks and Long Short-Term Memory • Combined multiple neural network architectures to extract and learn significant features from the input data • Designed a heart disease prediction model using Hybrid Deep Neural Network which combined both CNN and LSTM with additional dense layers • Model was evaluated on two publicly available heart disease datasets, including the Cleveland Heart Disease dataset, and a large public heart disease dataset (Switzerland + Cleveland + Statlog + Hungarian + Long Beach VA). • Proposed model achieved an accuracy of 98.86

Table 1.1: Literature Review

Chapter 2

Data Analysis

2.1 Introduction

This section provides the inclusive details of the strategy used to anticipate the manifestation of cardiovascular conditions. Many steps are taken to achieve the most optimal model for the Cardiovascular disease prediction.

2.2 Data Set

Dataset is taken from Kaggle. It is a cardiovascular disease dataset containing 12 features and 70000 instances. This dataset features different aspects like smoking, alcohol intake, physical activity, gender, age etc. to indicate the presence of cardiovascular disease The dataset is in .csv format and is of 1 MB. The dataset is balanced.

2.2.1 Attributes

There are three types of features:

1. **Objective Feature**: Patient's demographics
2. **Examination Feature**: Results of medical examination
3. **Subjective Feature**: Information given by the patient (lifestyle)

Attribute Name	Feature Type	Value Type
Age	Objective Feature	int (days)
Height	Objective Feature	int (cm)
Weight	Objective Feature	float (kg)
Gender	Objective Feature	categorical code (1 - women, 2 - men)
Systolic blood pressure	Examination Feature	int
Diastolic blood pressure	Examination Feature	int
Cholesterol	Examination Feature	1: normal, 2: above normal, 3: well above normal
Glucose	Examination Feature	1: normal, 2: above normal, 3: well above normal
Smoking	Subjective Feature	binary
Alcohol intake	Subjective Feature	binary
Physical activity	Subjective Feature	binary
Presence or absence of cardiovascular disease	Target Variable	binary

Table 2.1: Attributes of the dataset

2.2.1.1 Age

Data for age is given in days instead of years. Aging has increased affects on heart. A lot of changes occur in heart and blood vessels with the growing age. Older people are more likely to suffer a heart attack, having a stroke or heart failure than younger people. Below figure shows the data of age given in days.

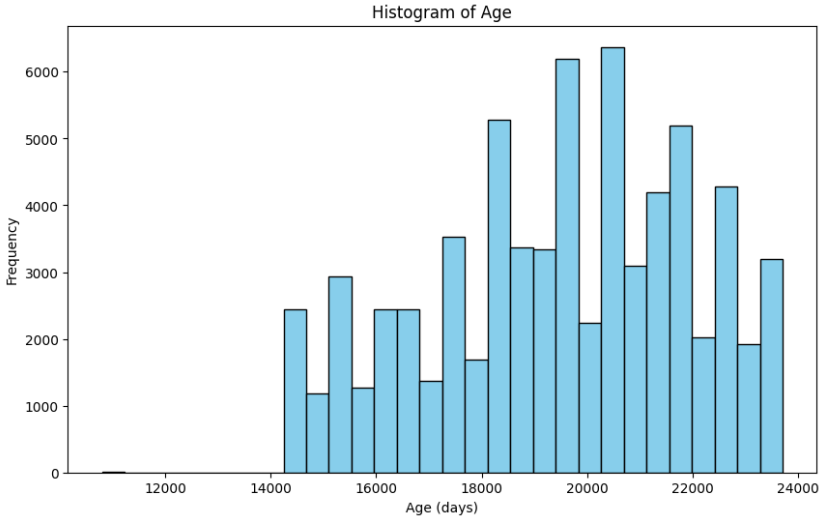


Figure 2.1: Histogram representing age in days

2.2.1.2 Height

Measurement of height is given in centimeters (cm). Height is one the factor that is related to medical conditions such as heart diseases. Recent researches show that height is being linked with risks of Cardiovascular Diseases (CVD). Taller people are less likely to have cardiovascular disease while the shorter people are at greater risk of CVD. Figure below represents the data of height.

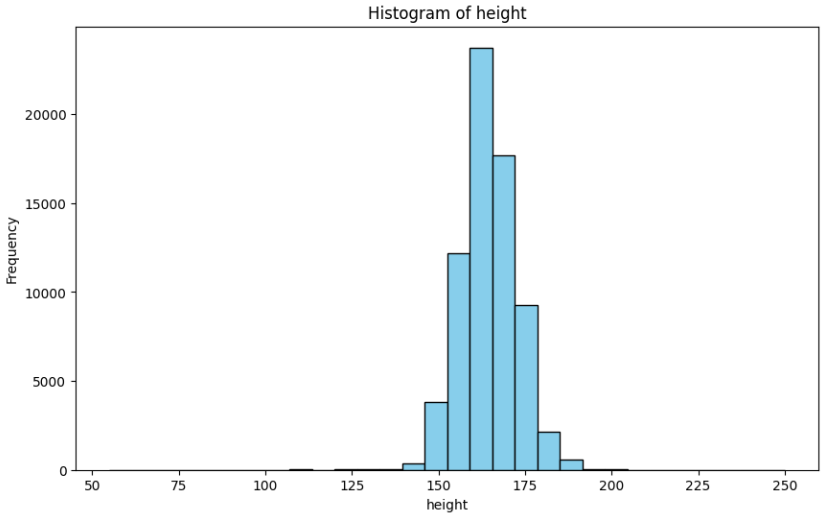


Figure 2.2: Histogram of height

2.2.1.3 Weight

Obesity is an increased risk factor for heart diseases. Excess weight leads to high blood pressure and also may form clots in blood vessels. You can see the data of weight visually which is given in kilograms (kg).

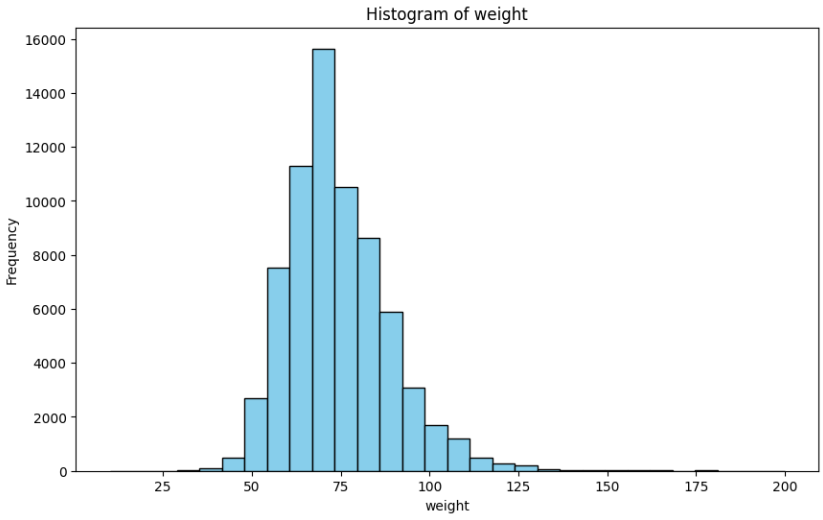


Figure 2.3: Histogram of weight

2.2.1.4 Gender

Cardiovascular disease is leading cause of morbidity and mortality worldwide. Although women generally have a lower prevalence of CVD than men. Number of studies has shown that after an acute cardiovascular (CV) event, women have a greater death rate and a worse prognosis [8].

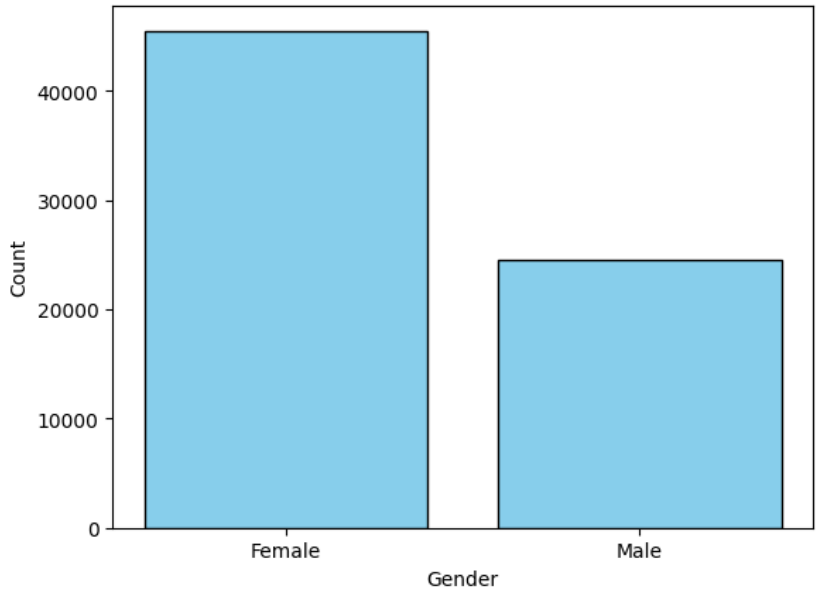


Figure 2.4: Gender

2.2.1.5 Systolic and diastolic blood pressure

Elevation of systolic blood pressure predicts the risk of cardiovascular disease better than increases in diastolic blood pressure. We plotted the values of ap_hi and ap_lo using Matplotlib to clearly visualize the abnormality of values and identify the outliers. The figure below clearly shows the values exceeding the limit.

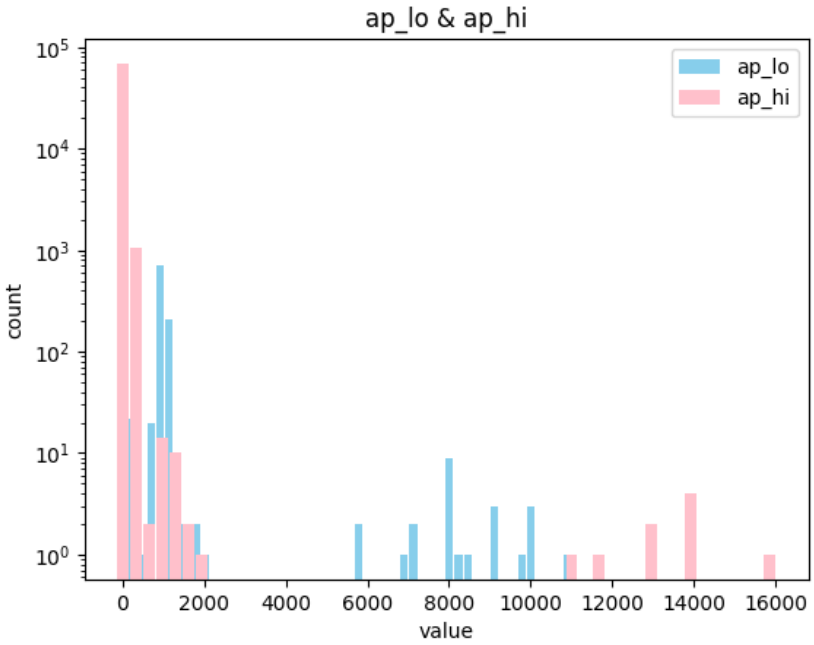


Figure 2.5: ap_hi and ap_lo

2.2.1.6 Cholesterol

Healthy cells are essential for human body to prevent from diseases. Human body needs cholesterol to build healthy cells but excess of cholesterol increases the risk of heart diseases. Figure below represents the cholesterol level as 0, 1, and 2 where 1 represents normal, 2 represents above normal, and 3 represents well above normal.

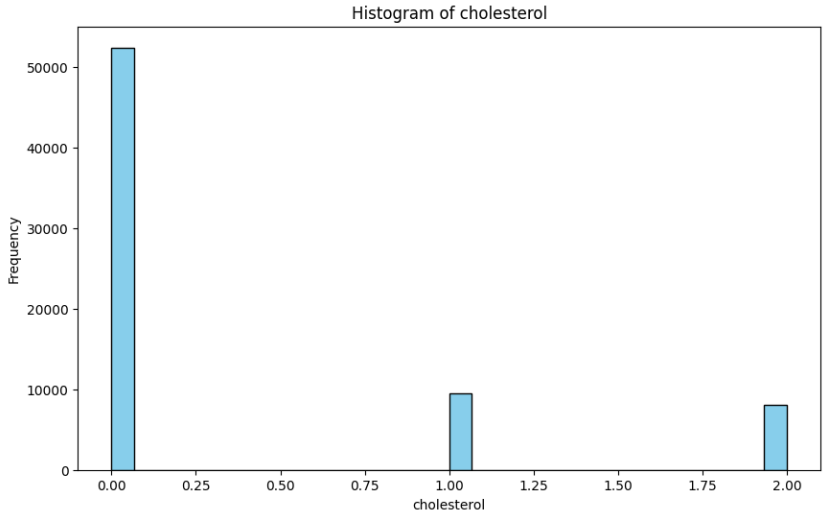


Figure 2.6: Cholesterol

2.2.1.7 Glucose

People can have serious heart problems if they have high sugar level that can damage the blood vessels. Figure below is representing the glucose level where 1 represents normal, 2 represents above normal, and 3 represents well above normal.

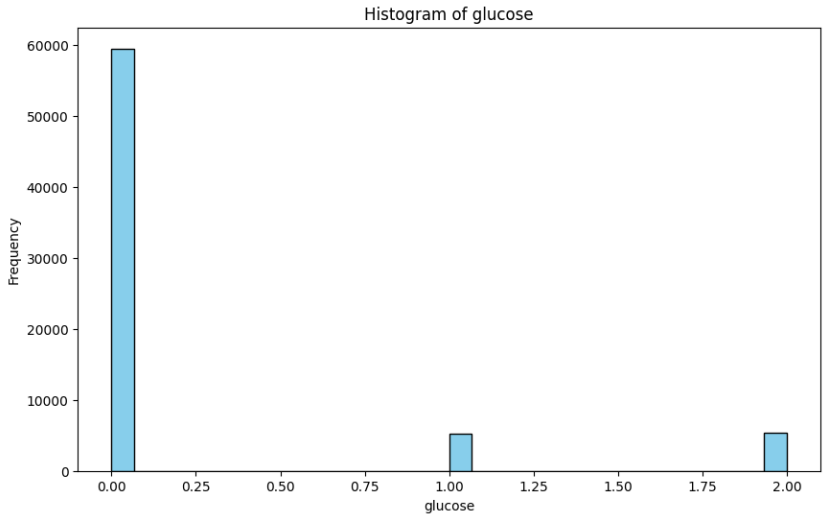


Figure 2.7: Glucose

2.2.1.8 Smoking

Smoking has severe affects on health. It increases the risk of developing the Cardiovascular disease and many other health conditions.

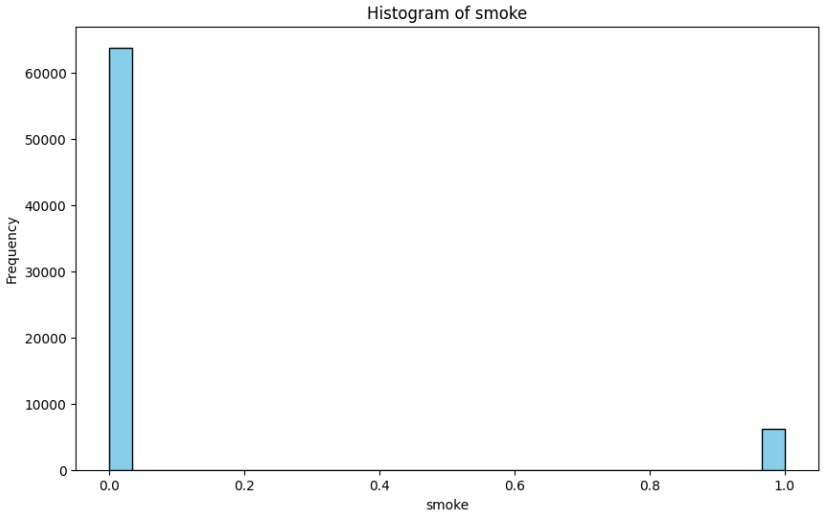


Figure 2.8: Smoking

2.2.1.9 Alcohol Intake

Heart failures and strokes are increasing day by day. Alcohol consumption has adverse effects on human health leading to different diseases including cardiovascular diseases.

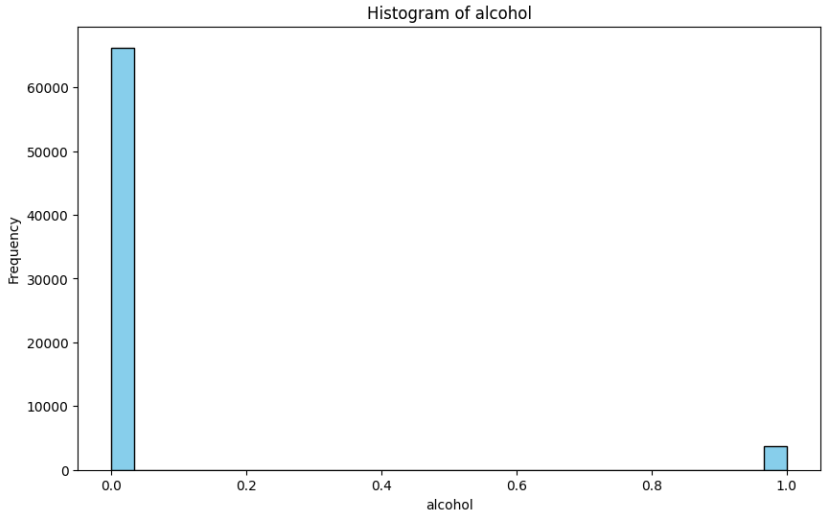


Figure 2.9: Alcohol Consumption

2.2.1.10 Physical Activity

Exercising or regular physical activity prevents from high blood pressure. It also lowers the blood cholesterol level reducing the risks of cardiovascular diseases.

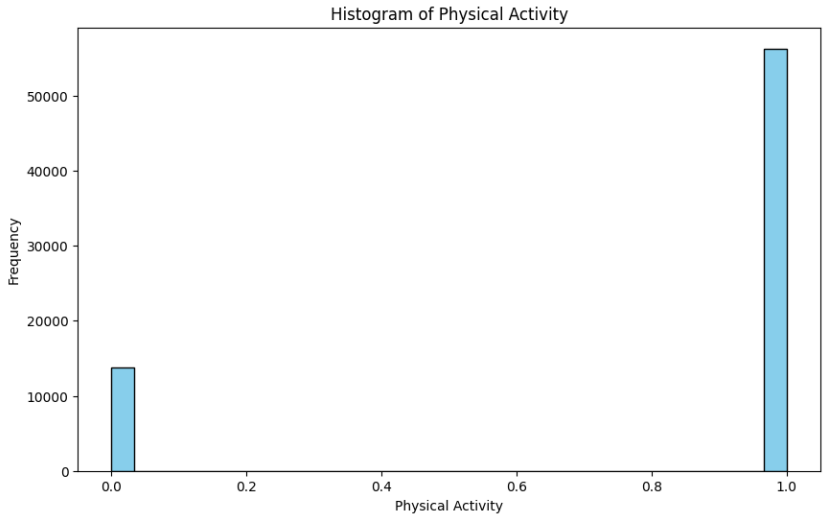


Figure 2.10: Physical Activity

2.2.1.11 Presence or absence of Cardiovascular disease

This is the target variable that indicates the presence or absence of cardiovascular disease. In the below figure, histogram clearly shows that the dataset is balanced.

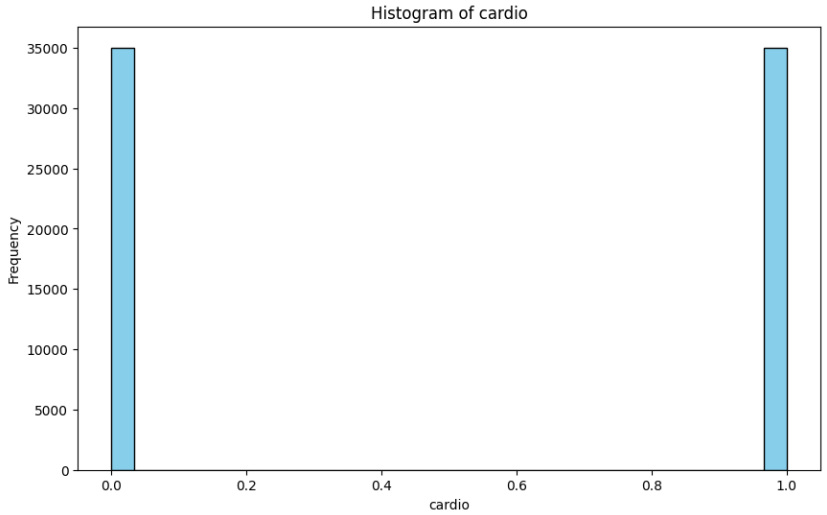


Figure 2.11: Presence or absence of CVD

2.3 Data Analysis

2.3.1 Data Cleansing

Drop is one of the main functions used to clean data.We can drop specified labels from rows or columns by using drop() by mentioning corresponding axis, index or column names, level when using multi index labels on different levels [9].

This dataset has two unnamed columns ”Column 1” and ”id”. Since in real time processing they are not needed for analysis so it was necessary to drop the columns in order to get the best performing models.

Dropping

```
[ ]  datadrop= data.drop(columns=['Column 1', 'id'])
```

Figure 2.12: Python code for data cleansing

2.3.2 Train Test Split

Train test split is a validation technique that allows to simulate how your model performs on an unseen or new data by dividing the entire dataset into training and testing subsets.

Here is the working of train test split [10]:

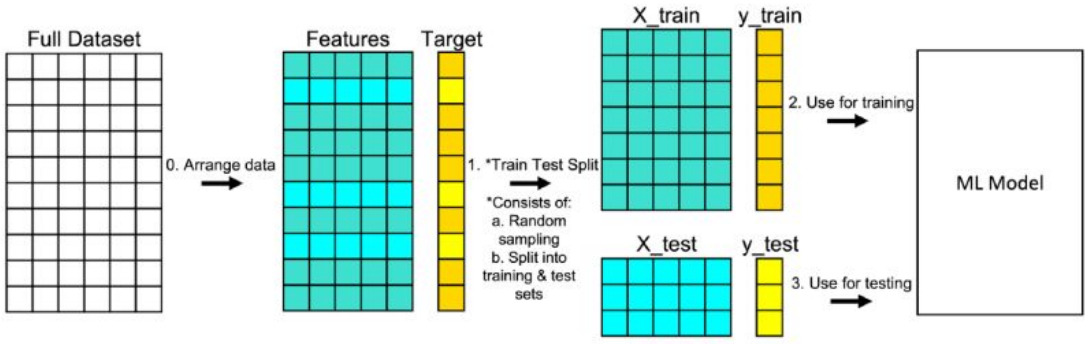


Figure 2.13: Train Test split

We splitted the data into X train, X test, y train and y test using train test split. Here, X contains all the columns of the dataframe ”datadrop” except the column cardio and y the target variable contains the column cardio which we want to predict.

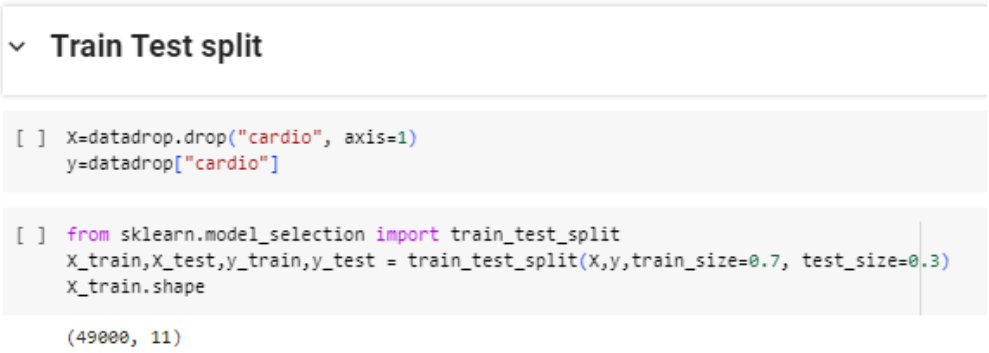


Figure 2.14: Python code for Train test split

2.3.3 Overfitting

Overfitting occurs when machine learning model tries to cover all data points or more than the required data points present in the given dataset.

Since some of the data values are abnormally high so the machine learning algorithms were giving 100% accuracy. So to avoid overfitting, we used StandardScaler that will standardize the input data.

2.3.4 Scaling

StandardScaler is used to standardize the input data so that the data points have a balanced scale. Standardization transforms the data such that the mean of each feature becomes zero (centered at zero), and the standard deviation becomes one.

Some of the feature values of the dataset were fluctuating within their ranges so here we used StandardScaler.

Scaling

```
[ ]  import numpy as np
     from sklearn.preprocessing import StandardScaler

     # Initialize StandardScaler
     scaler = StandardScaler()

     # Fit and transform the training data
     X_train_scaled = scaler.fit_transform(X_train)

     # Transform the testing data using the same scaler
     X_test_scaled = scaler.transform(X_test)
```

Figure 2.15: Python Code for Scaling

Since the standardization affect the accuracy of the models so later on we will discuss the results we got before and after scaling.

2.3.5 Packages Used

The main python packages used in this project are:

Pandas	Data frames
Numpy	Numerical calculations and Arrays
Matplotlib	Visualization of Data
Sklearn	Importing Ml Models
Tensorflow.keras.models	Importing DL Neural Network layers

Table 2.2: Packages

Chapter 3

Machine and Deep Learning

3.1 Machine Learning Techniques

This section outlines the machine learning techniques applied to the given dataset to attain the outperforming approach.

3.1.1 Decision Tree

Decision tree is a supervised machine learning technique that can be used for both regression and classification tasks. It is a non-parametric algorithm. It is a tree like structure graphically presenting all feasible solutions for a given problem.

It consists of internal nodes representing the features of a dataset, branches representing the decision rules and each leaf node representing the outcome.

There are two nodes in a decision tree namely Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

Among many other algorithms, it is mostly used due to:

- Easily understandable due to tree-like structure
- They closely mimics the way humans make decisions making it easy to understand

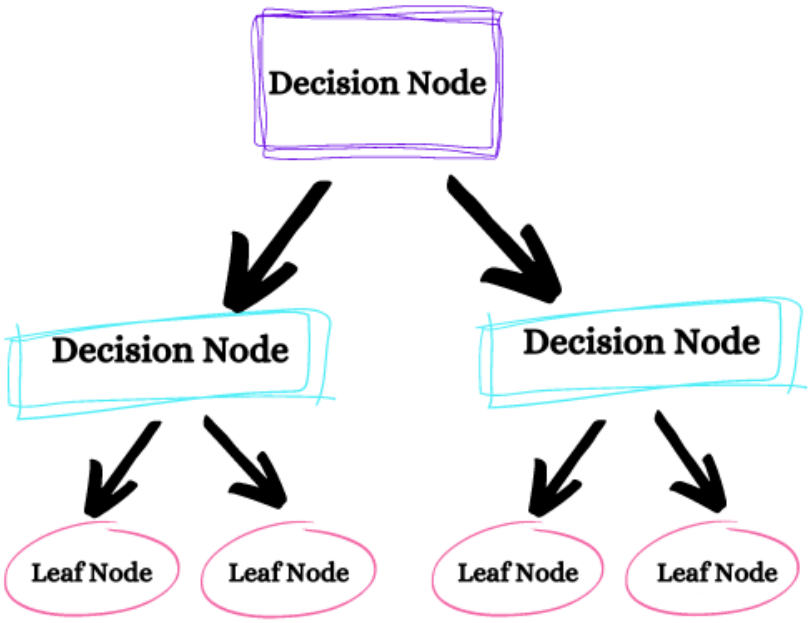


Figure 3.1: General structure of Decision Tree

3.1.1.1 Algorithm

1. Start with the root node having complete dataset.
2. Using Attribute Selection Measure (ASM), identify the best feature in dataset.
3. Divide the root node into subsets that hold the possible values for the best feature.
4. Generate the decision tree node containing the best feature.
5. Repeatedly make decision tree nodes using subsets in the step 3. Continue the process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

3.1.2 Random Forest

Several trees put together form a forest (in real life); several decision trees form a random forest (in machine learning).

Random Forest is a supervised learning technique. It can be used for both regression and classification problems. It is based on the concept of ensemble learning, which combines multiple models to make prediction rather than individual model.

3.1.2.1 Algorithm

Given below is the random forest algorithm.

1. Select random K data points from the training set.
2. Build the decision trees associated with the selected data points (Subsets).
3. Choose the number N for decision trees that you want to build.
4. Repeat Step 1 and 2.
5. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

3.1.3 K-Nearest Neighbors

K-Nearest Neighbors also known as KNN is supervised learning technique. It is a non-parametric and one of the simplest machine learning technique. It does not make assumptions on the underlying data.

This algorithm can be used for both regression and classification problems but mostly used for classification problems.

3.1.3.1 Algorithm

K-Nearest Neighbors algorithm predicts the values of new data points using the feature similarity. It checks the similarity between the new data point and the training data set and puts the new data point to the most similar category. Here are the following working steps of the algorithm:

1. During the first step, load the training as well as testing data.
2. Choose the value of K; the nearest data points. $K = (\text{Total Instances})^{\frac{1}{2}}$
3. Calculate the distance basically a similarity between test data and each row of training data using Euclidean distance.
4. Choose the top K nearest neighbors as per Euclidean distance.
5. Assign a class to the test point based on the most frequently occurring class.
6. Our model is ready.

3.1.4 Logistic Regression

Logistic regression is one of the popular supervised learning algorithm. It is used for solving classification tasks with a goal to predict the probability that an instance belongs to a given class or not. It allocates probabilities to discrete outcomes using the Sigmoid function, which converts numerical results into probabilistic values that lie between 0 and 1. The dependent variable is categorical in nature.

Logistic regression is quiet similar to linear regression but logistic regression is used for classification problems, whereas linear regression for regression problems.

3.1.4.1 Sigmoid Function

The name "logistic regression" is derived from the concept of logistic function. Logistic function is also known as sigmoid function which is used to map the predicted values of probabilities.

The value of the logistic regression lies between 0 and 1, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

3.1.5 Naive Bayes

Naive Bayes is a supervised learning technique used for classification tasks. It is mostly used for text classification. It is also known as probabilistic classifier since it is based on Bayes theorem.

The essential Naïve Bayes assumption is that each feature makes an independent equal contribution to the outcome.

3.1.5.1 Bayes Theorem

Bayes theorem is a rule used to determine the conditional probability of events. Essentially, the Bayes' theorem describes the probability of an event based on prior knowledge of the conditions that might be relevant to the event [11].

Mathematically,

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}} \tag{3.1}$$

Where:

- $P(A|B)$ – the probability of event A occurring, given event B has occurred
- $P(A|B)$ – the probability of event B occurring, given event A has occurred
- P(A) – the probability of event A
- P(B) – the probability of event B

3.1.5.2 Algorithm

1. Convert the given dataset into frequency tables.
2. Generate Likelihood table by finding the probabilities of given features.
3. Now, use Bayes theorem to calculate the posterior probability.

3.1.5.3 Applications

Some of the most popular examples of Naive Bayes are:

- Spam filtration
- Classifying articles
- Sentimental analysis

3.2 Deep Learning Models

Deep learning is a subset of machine learning that is based on artificial neural networks to mimic learning capability of humans. Here are a few deep learning models we applied on the taken dataset.

3.2.1 Convolutional Neural Network (CNN)

One of the popular deep neural network is Convolutional Neural Network. CNN is particularly used for image recognition and tasks processing. It consists of three layers including convolutional layers, pooling layers, and fully connected layers.

The figure below shows the structure of convolutional neural network [12].

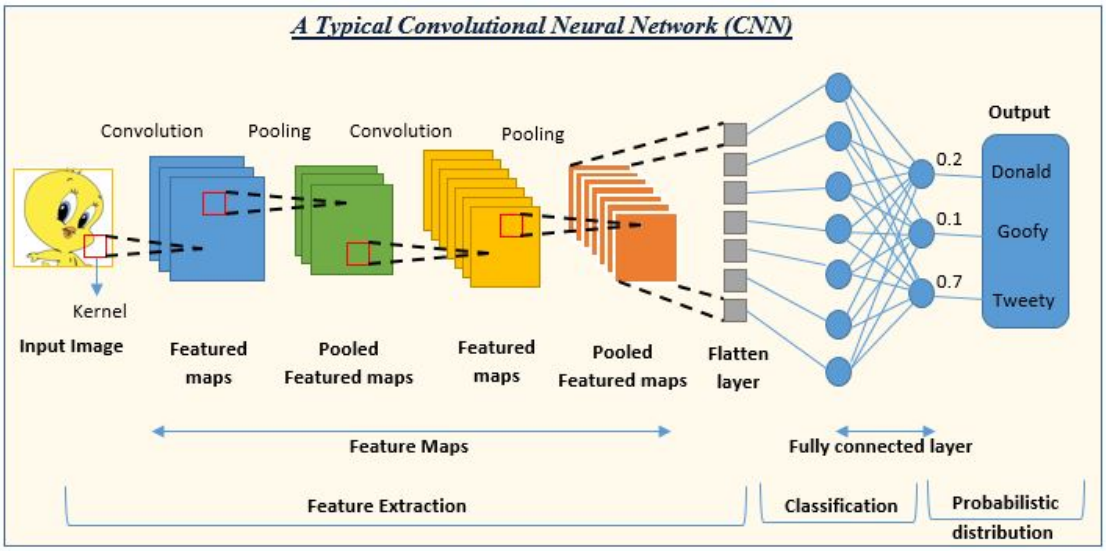


Figure 3.2: Convolutional Neural Network

3.2.1.1 Convolutional Layers

The convolutional layer is the building block of the CNN. The key components of CNN are input data, filters, and feature maps.

The convolutional layer computes a dot product between the filter value and the image pixel values, and the matrix formed by sliding the filter over the image is called the Convolved Feature, Activation Map, or Feature Map [13]. The sliding size of kernel is called a stride.

If we have an input of size W x W x D and D_{out} number of kernels with a spatial size of F with stride S and amount of padding P, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

3.2.1.2 Pooling Layers

Pooling layer reduces the dimensionality of the input, there by decreasing the number of parameters. Pooling filter does not possess weights. Max pooling is a common pooling operation used which select the maximum value from a group of neighboring pixels.

If we have an activation map of size W x W x D, a pooling kernel of spatial size F, and stride S, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F}{S} + 1$$

This will yield an output volume of size W_{out}x W_{out} x D.

3.2.1.3 Fully Connected Layers

This layer connects the information extracted from preceding layers to the output layer and classifies the input into desirable output as seen in regular fully connected neural network (FCNN). This is why it can be computed as usual by a matrix multiplication followed by a bias effect.

The fully connected layers are used to flatten the 2D spatial structure of the data into a 1D vector and process this data for tasks like classification.

The fully connected layer helps to map the representation between the input and the output [14]

3.2.1.4 Limitations

1. Large datasets are required
2. Time consuming
3. High computational requirements
4. Sensitive to adversarial attacks

5. Hard to interpret

3.2.2 Artificial Neural Network(ANN)

Artificial Neural Networks contain artificial neurons also known as units. These units are arranged in a series of layers that together constitute the whole Artificial Neural Network in a system. The number of units in a layer can vary from few dozen units to millions of units depending on how the complex neural networks will be required to learn the hidden patterns in the dataset [15].

3.2.2.1 Algorithm

Artificial Neural Network has an input layer, an output layer as well as hidden layers.

- The input layer receives data from the outside world which the neural network needs to analyze or learn about.
- Then this data passes through one or multiple hidden layers that transform the input into data that is valuable for the output layer.
- The output layer provides an output in the form of a response of the Artificial Neural Networks to input data provided.

3.2.2.2 Limitations

- Lack of transparency
- Difficulties in introducing problems to artificial neural networks

3.2.2.3 Applications

- Image recognition
- Speech recognition

- Machine translation
- Medical diagnosis

3.2.3 Deep Belief Network(DBN)

Deep belief networks (DBNs) are a type of deep learning algorithm that addresses the problems associated with classic neural networks. DBN is an unsupervised probabilistic deep learning model.

DBN is a hybrid generative graphical model. The top two layers have no direction. The layers above have directed links to lower layers.

Several Restricted Boltzmann Machines together make a Deep Belief Networks.

3.2.3.1 Restricted Boltzmann Machine

Restricted Boltzmann Machine is a probabilistic, unsupervised, generative deep machine learning algorithm.

3.2.3.2 How does DBN work?

Working of deep belief networks is as follows:

- We'll pre-train the DBN using the Greedy learning algorithm. For learning the top-down generative weights-the greedy learning method that employs a layer-by-layer approach. The relationship between variables in one layer and variables in the layer above is determined by these generative weights.
- On the top two hidden layers, we run numerous steps of Gibbs sampling in DBN. The top two hidden layers define the RBM thus, this stage is effectively extracting a sample from it.
- Then generate a sample from the visible units using a single pass of ancestral sampling through the rest of the model.

- We'll use a single bottom-up pass to infer the values of the latent variables in each layer. In the bottom layer, greedy pretraining begins with an observed data vector. It then oppositely fine-tunes the generative weights.

3.2.4 Recurrent Neural Network (RNN)

A Deep Learning approach for modelling sequential data is Recurrent Neural Networks. In recurrent neural networks, the output from the previous step is fed as input to the current step. For tasks that involve sequential inputs, such as speech and language, it is often better to use RNNs.

In an NLP problem, if you want to predict the next word in a sentence it is important to know the words before it. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being depended on the previous computations. Another way to think about RNNs is that they have a "memory" which captures information about what has been calculated so far.

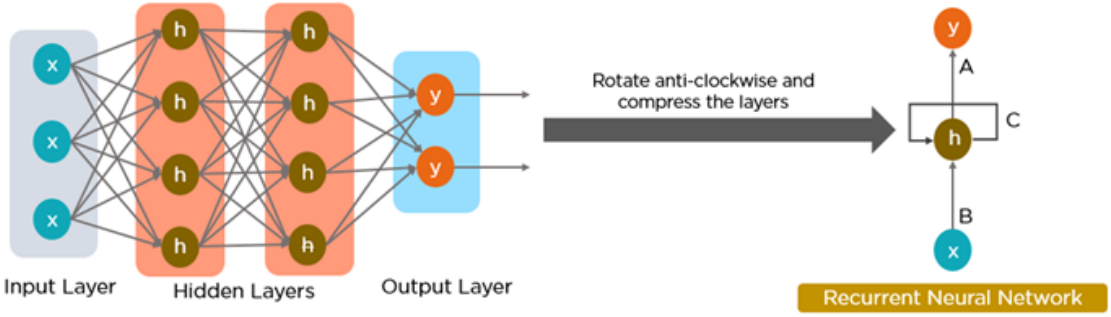
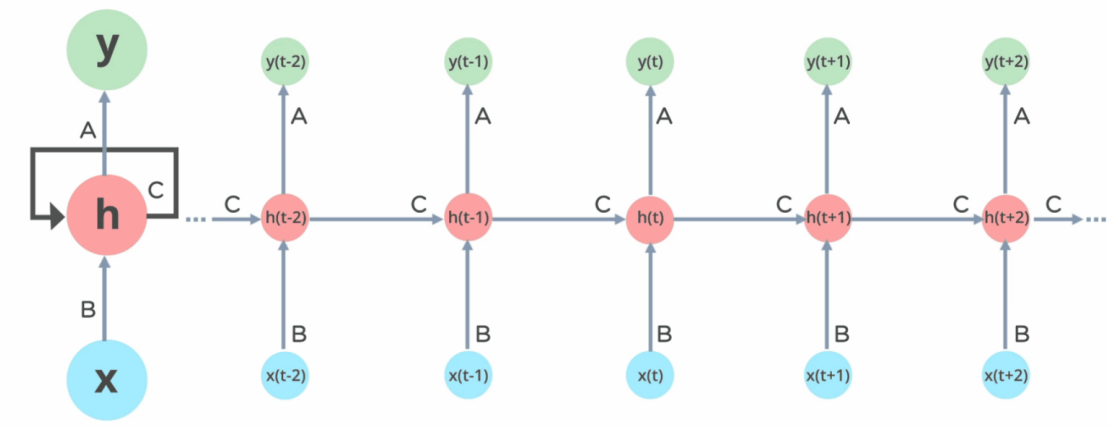


Figure 3.3: Recurrent Neural Network

3.2.4.1 How does recurrent neural networks work?

The information in recurrent neural networks cycles through a loop to the middle hidden layer.



The input layer x receives and processes the neural network's input before passing it on to the middle layer.

Multiple hidden layers can be found in the middle layer h, each with its own activation functions, weights, and biases. You can utilize a recurrent neural network if the various parameters of different hidden layers are not impacted by the preceding layer, i.e. There is no memory in the neural network.

3.2.4.2 Activation Functions

A neuron's activation function dictates whether it should be turned on or off. Nonlinear functions usually transform a neuron's output to a number between 0 and 1 or -1 and 1.

The following are some of the most commonly utilized functions:

- **Sigmoid**: The formula
$$g(z) = \frac{1}{1 + e^{-z}}$$
is used to express this.
- **Tanh**: The formula
$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$
is used to express this.

- **ReLU**: The formula g(z) = max(0 , z) is used to express this.

3.2.4.3 Limitations

1. Vanishing and exploding gradients
2. Computational complexity
3. Lack of parallelism
4. Difficulty in interpreting the output
5. Difficulty in capturing long term dependencies

3.2.5 Gated Recurrent Network (GRU)

GRU stands for Gated Recurrent Unit, which is a type of recurrent neural network (RNN). Architecture of GRU is similar to LSTM (Long Short-Term Memory).

The primary distinction between GRU and LSTM is the way they handle the memory cell state. In LSTM, the memory cell state is maintained separately from the hidden state and is updated using three gates: the input gate, output gate, and forget gate. A "candidate activation vector," which is updated via the reset and update gates, takes the place of the memory cell state in GRU.

The reset gate determines how much of the previous hidden state to ignore, while the update gate determines how much of the candidate activation vector to incorporate into the new hidden state.

The figure below shows the full architecture of GRU [16]:

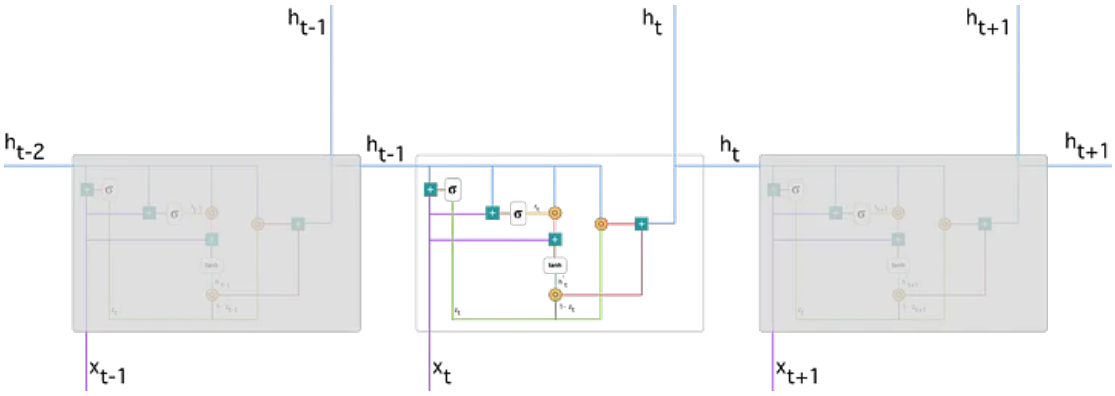


Figure 3.4: Recurrent neural network with Gated Recurrent Unit

3.2.5.1 How does GRU work?

To solve the vanishing gradient problem of a standard RNN, GRU uses, so-called, update gate and reset gate. Basically, these are two vectors which determine what data should be sent to the output. Their unique quality lies in their ability to be trained to keep data from long ago, without erasing it through time or remove data which is irrelevant to the prediction. Here is a detailed version of GRU [16]:

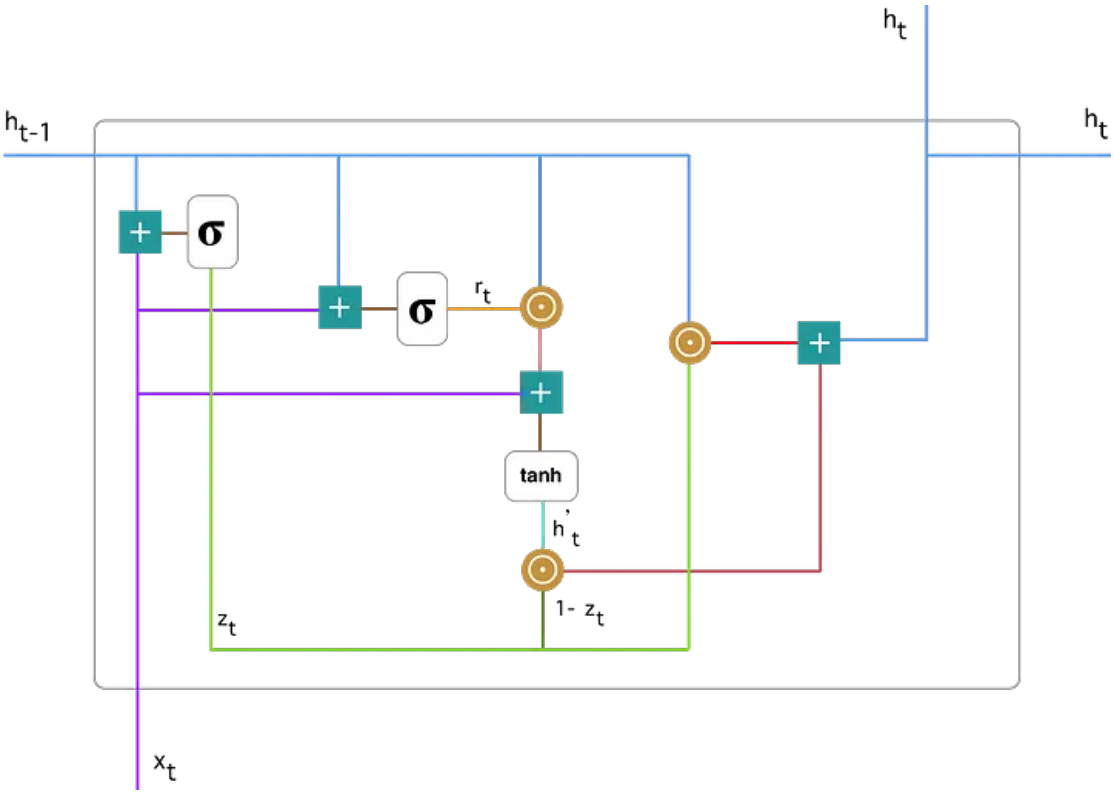


Figure 3.5: Gated Recurrent Unit

Notations used in this figure are:

Figure 3.6: Notations

3.2.5.2 Update Gate

Using the following formula, We compute the update gate z_t for time step t:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

When x_t is plugged into the network unit, it is multiplied by its own weight W(z). The same goes for $h_{(t-1)}$ which holds the data for the previous t-1 units and

is multiplied by its own weight U(z). After adding the both results, a sigmoid activation function is applied to squash the result between 0 and 1.

3.2.5.3 Reset Gate

The model uses reset gate to decide how much of the past data to forget. To calculate it, we use:

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

3.2.5.4 Current Memory content

We introduce a new memory content which will use the reset gate to store the relevant data from the past. The following formula is used to compute it:

$$h'_t = tanh(Wx_t + r_t \odot Uh_{t-1})$$

1. Multiply the input x_t with a weight W and $h_{(t-1)}$ with a weight U.
2. Calculate the Hadamard (element-wise) product between the reset gate r_t and $Uh_{(t-1)}$. That will determine what to remove from the previous time steps.
3. Sum up the results of step 1 and 2.
4. Apply the nonlinear activation function tanh.

3.2.5.5 Final Memory at current time step

Using the update gate at the final step, the network calculates h_t — vector which holds data for the current unit and passes it down to the network. It determines what to collect from the current memory content — h_t and what from the previous steps — $h_{(t-1)}$. That is done as follows:

$$h_t = z_t \odot h_{(t-1)} + (1 - z_t) \odot h'_t$$

1. Apply element-wise multiplication to the update gate z_t and $h_{(t-1)}$.
2. Apply element-wise multiplication to $(1 - z_t)$ and h_t.
3. Sum the results from step 1 and 2 [16].

3.2.5.6 Limitations

1. More prone to overfitting
2. May not perform well
3. Require careful tuning of hyperparameters

3.2.6 AlexNet

3.2.6.1 Architecture

The architecture consists of eight layers: five convolutional layers and three fully-connected layers. But this isn't what makes AlexNet special; these are some of the features used that are new approaches to convolutional neural networks:

1. **ReLU Nonlinearity**

   AlexNet uses Rectified Linear Units (ReLU) instead of the tanh function, which was standard at the time.

2. **Multiple GPUs**

   Back in the day, GPUs were still rolling around with 3 gigabytes of memory. This was especially bad because the training set had 1.2 million images. AlexNet allows for multi-GPU training by putting half of the model's neurons on one GPU and the other half on another GPU.

3. **Overlapping Pooling**

   CNNs traditionally "pool" outputs of neighboring groups of neurons with no overlapping. However, when the authors introduced overlap, they saw a reduction in error by about 0.5% and found that models with overlapping pooling generally find it harder to overfit.

4. **Dropout**

   During dropout, a neuron is eliminated from the neural network with a probability of 0.5. A neuron that is dropped does not make any contribution to either forward or backward propagation.

This table below shows the architecture of alexnet [17]:

Layer	# filters / neurons	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-	-	-	227 x 227 x 3	-
Conv 1	96	11 x 11	4	-	55 x 55 x 96	ReLU
Max Pool 1	-	3 x 3	2	-	27 x 27 x 96	-
Conv 2	256	5 x 5	1	2	27 x 27 x 256	ReLU
Max Pool 2	-	3 x 3	2	-	13 x 13 x 256	-
Conv 3	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 4	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 5	256	3 x 3	1	1	13 x 13 x 256	ReLU
Max Pool 3	-	3 x 3	2	-	6 x 6 x 256	-
Dropout 1	rate = 0.5	-	-	-	6 x 6 x 256	-

Figure 3.7: AlexNet Summary

3.2.6.2 Limitations

- Large size
- High computational cost
- Relatively slow inference time

Chapter 4

Results and Discussions

4.1 Terminologies Used

4.1.1 Accuracy

Accuracy is widely known performance metric for machine learning algorithms. It measures how often a machine learning model predicts the correct output.

Mathematically,

$$Accuracy = \frac{\text{Correct Predictions}}{\text{All predictions}}$$

4.1.2 Precision

Precision is a metric that measures how frequently a machine learning model predicts the positive class.

Mathematically,

$$Precision = \frac{\text{True positives}}{\text{True positives} + \text{ False positives}}$$

4.1.3 Recall

Recall is a performance metric that how often a machine learning model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset.

Mathematically,

$$Recall = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

4.1.4 F1 Score

The F1 score ranges from 0 to 1. It is known as balancing of precision and recall.

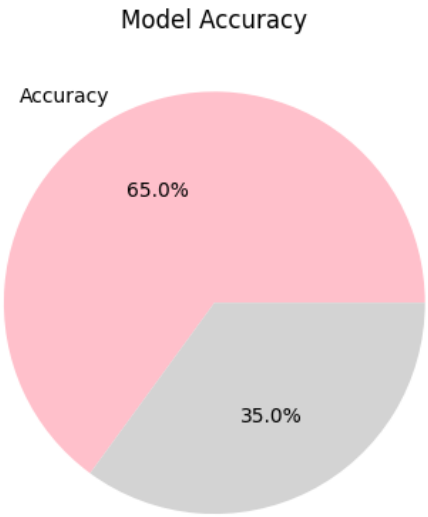
Mathematically,

$$F1Score = 2 * \frac{precision * recall}{precision + recall}$$

4.2 Machine Learning Techniques

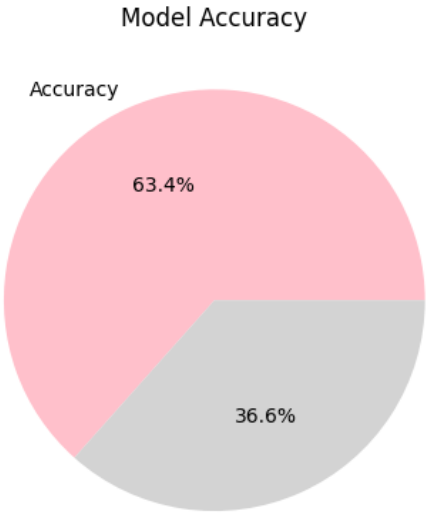
4.2.1 K-Nearest Neighbors

K-Nearest Neighbors achieved an accuracy of 65.0% on the test data.



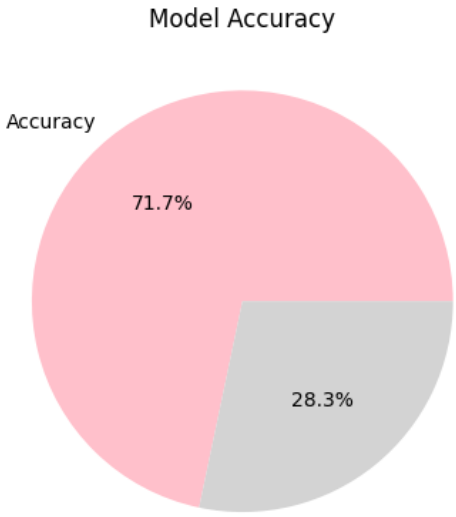
4.3 Decision Tree

Decision tree got an accuracy of 63.4% on the test data.



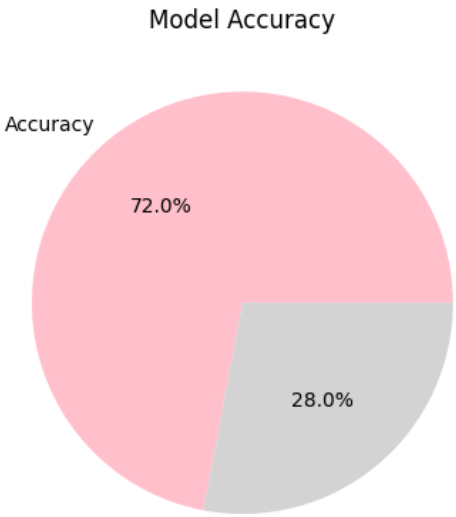
4.4 Random Forest

Random forest achieved an accuracy of 71.7%. Due to its ensemble nature, random forest performed 8% better than the decision tree model.



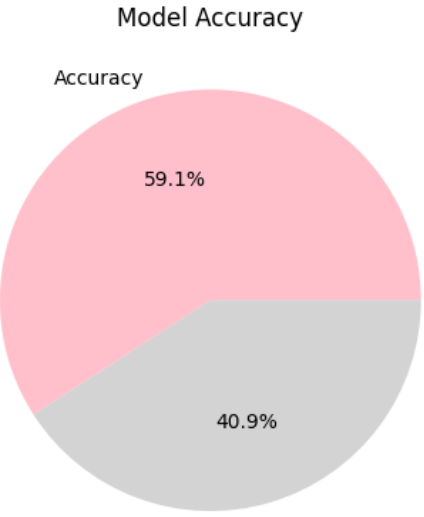
4.5 Logistic Regression

Logistic regression achieved an accuracy of 72.0% more than all other machine learning techniques. It showed reliable performance for predicting cardiovascular disease.



4.6 Naive Bayes

Naive bayes showed an accuracy of 59.1%. Naive bayes often efficiently predict diseases but may not capture complex relationships between features.



4.6.1 Results without Scaling

The table shows the results of machine learning techniques without scaling.

Techniques	Accuracy	Precision	F1 Score	Recall Score
KNN	0.68131	0.67803	0.68661	0.69541
Decision Tree	0.61421	0.62145	0.63172	0.62636
Random Forest	0.69169	0.69741	0.69965	0.70190
Logistic Regression	0.69922	0.68055	0.69970	0.71996
Naive Bayes	0.57550	0.39279	0.48793	0.64390

Table 4.1: Results

4.6.2 Results with Scaling

The table shows the results of machine learning techniques with scaling. Accuracy of all techniques almost increased by 2% with scaling but accuracy of KNN decreased by 4%.

Techniques	Accuracy	Precision	F1 Score	Recall Score
KNN	0.6495	0.6298	0.6419	0.6544
Decision Tree	0.6342	0.6310	0.6324	0.6339
Random Forest	0.7170	0.7061	0.7133	0.7208
Logistic Regression	0.7203	0.6847	0.7095	0.7362
Naive Bayes	0.5912	0.3970	0.4921	0.6471

Table 4.2: Results

The comparison between accuracy, precision, recall and F1 score of all the machine learning techniques applied on the dataset is obvious. This figure shows the comparison of results with scaling. Logistic regression got higher accuracy of 72.03% among all the machine learning techniques and naive bayes got least accuracy of 59.12%.

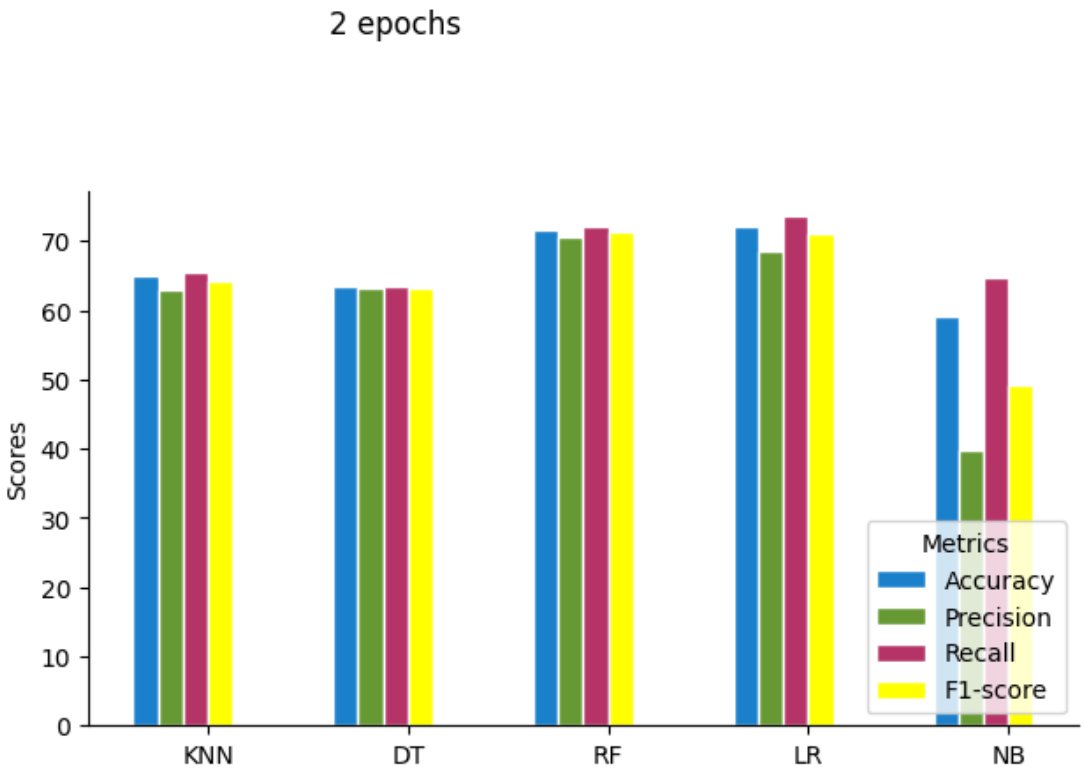
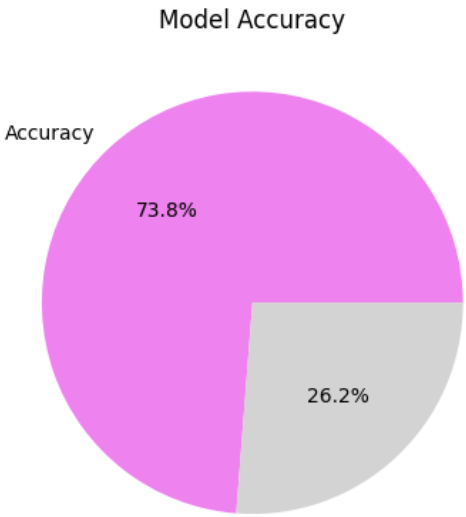


Figure 4.1: Performance comparison of machine learning techniques

4.7 Deep Learning Models

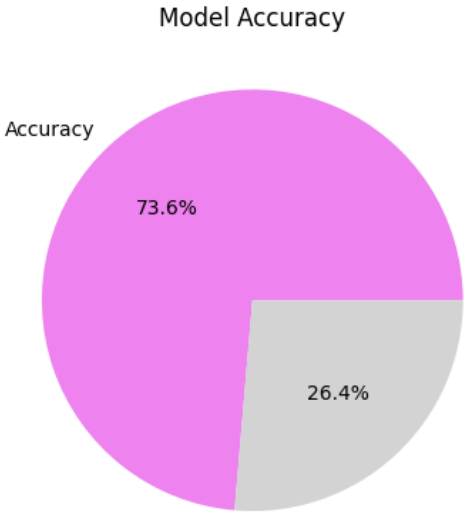
4.7.1 Convolutional Neural Network

Convolutional neural network achieved an accuracy of 73.8% due to its deep architecture.



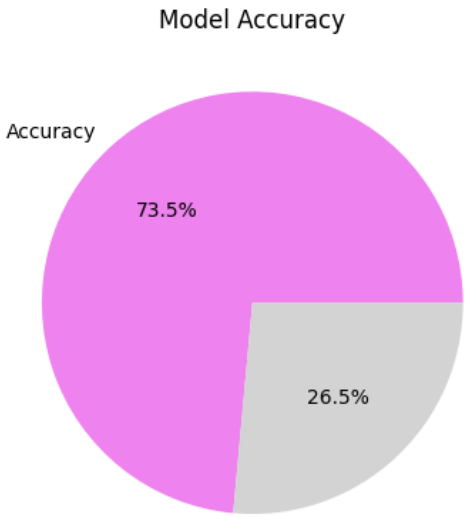
4.7.2 Deep Belief Network

Deep belief network achieved an accuracy of 73.6%.



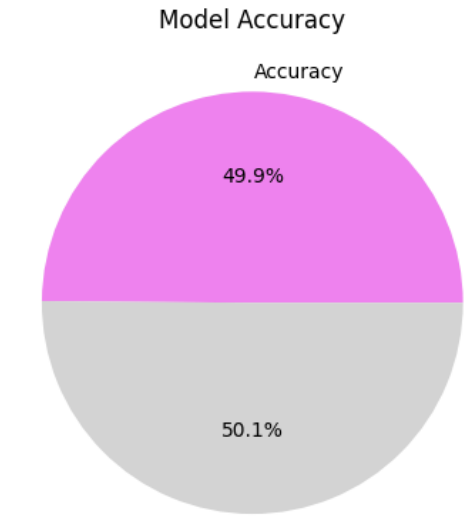
4.7.3 Artificial Neural Network

Artificial neural network achieved an accuracy of 73.5%



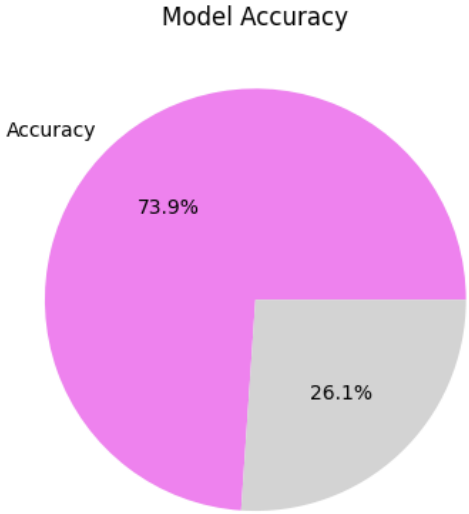
4.7.4 Recurrent Neural Network

Recurrent neural network got an accuracy of 49.9% showing its limited capability to accurately predict the disease.



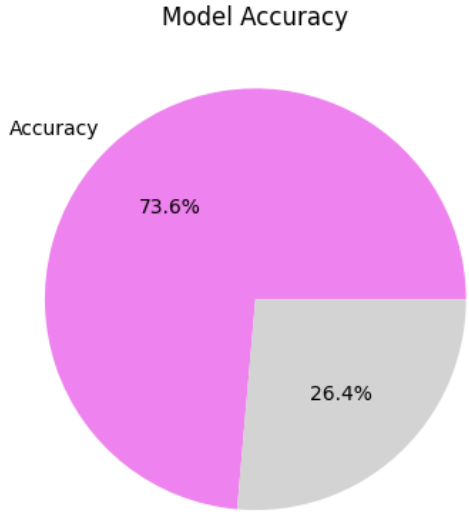
4.7.5 Gated Recurrent Network

Gated recurrent network achieved an highest accuracy of 73.9% among all other deep learning models.



4.7.6 AlexNet

AlexNet achieved an accuracy of 73.6% as shown in figure.



Nearly all deep learning models achieved similar accuracy except for the RNN.

4.7.7 Results without Scaling

The table shows the results of deep learning models without scaling.

Techniques	Accuracy	Precision	F1 Score	Recall Score
CNN	0.6984	0.7638	0.6616	0.5834
DBN	0.7093	0.7470	0.6999	0.6584
ANN	0.6930	0.6945	0.7210	0.7075
RNN	0.7240	0.7335	0.7228	0.7125
GRU	0.7036	0.7935	0.6659	0.5737
AlexNet	0.7270	0.73283	0.7262	0.7270

Table 4.3: Results

4.7.8 Results with Scaling

The table below shows the results of deep learning models with scaling. Accuracy of all techniques almost increased by 4% with scaling but accuracy of RNN decreased by almost 22.53%.

Techniques	Accuracy	Precision	F1 Score	Recall Score
CNN	0.7377	0.7445	0.7330	0.7219
DBN	0.7364	0.7545	0.7257	0.6991
ANN	0.7354	0.7455	0.7289	0.7130
RNN	0.4987	0.4988	0.6655	0.9999
GRU	0.7392	0.7625	0.7260	0.6928
AlexNet	0.7361	0.7581	0.7233	0.6916

Table 4.4: Results

The comparison between accuracy, precision, recall and F1 score of all the deep learning models applied on the dataset is obvious. This figure shows the comparison of results with scaling. Gated Recurrent Unit (GRU) got higher accuracy of

73.92% among all the deep learning models and Recurrent Neural Network (RNN) got least accuracy of 49.87%.

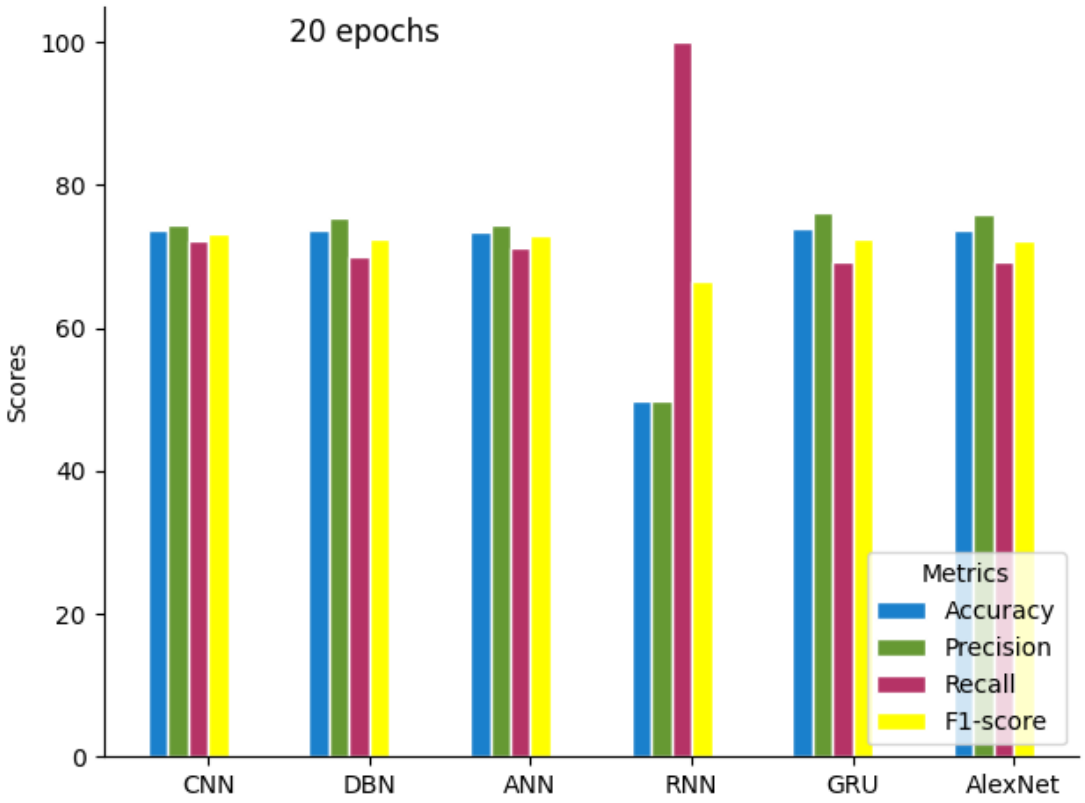


Figure 4.2: Performance comparison of deep learning models

4.8 Conclusion

It is necessary to diagnose the cardiovascular diseases at an early stage to decrease the mortality rate. Many algorithms are not able to correctly identify the cardiovascular disease. Even sometimes doctors are not able to identify them. In this project, we used many state-of-the-art algorithms to monitor their performance in predicting the cardiovascular disease. The reported accuracy of GRU is 73.92% on cardiovascular disease dataset taken from kaggle.

Bibliography

[1] Pronab Ghosh, Sami Azam, Mirjam Jonkman, Asif Karim, FM Javed Mehedi Shamrat, Eva Ignatious, Shahana Shultana, Abhijith Reddy Beeravolu, and Friso De Boer. Efficient prediction of cardiovascular disease using machine learning algorithms with relief and lasso feature selection techniques. *IEEE Access*, 9:19304–19326, 2021.

[2] Tulasi Krishna Sajja and Hemantha Kumar Kalluri. A deep learning method for prediction of cardiovascular disease using convolutional neural network. *Rev. d'Intelligence Artif.*, 34(5):601–606, 2020.

[3] Yong Li, Zihang He, Heng Wang, Bohan Li, Fengnan Li, Ying Gao, and Xiang Ye. Craftnet: a deep learning ensemble to diagnose cardiovascular diseases. *Biomedical Signal Processing and Control*, 62:102091, 2020.

[4] Amer M Johri, Krishna V Singh, Laura E Mantella, Luca Saba, Aditya Sharma, John R Laird, Kumar Utkarsh, Inder M Singh, Suneet Gupta, Manudeep S Kalra, et al. Deep learning artificial intelligence framework for multiclass coronary artery disease prediction using combination of conventional risk factors, carotid ultrasound, and intraplaque neovascularization. *Computers in Biology and Medicine*, 150:106018, 2022.

[5] Madhumita Pal, Smita Parija, Ganapati Panda, Kuldeep Dhama, and Ranjan K Mohapatra. Risk prediction of cardiovascular disease using machine learning classifiers. *Open Medicine*, 17(1):1100–1113, 2022.

[6] Shinthi Tasnim Himi, Natasha Tanzila Monalisa, MD Whaiduzzaman, Alistair Barros, and Mohammad Shorif Uddin. Medai: A smartwatch-based application framework for the prediction of common diseases using machine learning. *IEEE Access*, 11:12342–12359, 2023.

[7] Mana Saleh Al Reshan, Samina Amin, Muhammad Ali Zeb, Adel Sulaiman, Hani Alshahrani, and Asadullah Shaikh. A robust heart disease prediction system using hybrid deep neural networks. *IEEE Access*, 2023.

[8] Satyam Suman, Jakkula Pravalika, Pulluru Manjula, and Umar Farooq. Gender and cvd- does it really matters? *Current Problems in Cardiology*, 48(5):101604, 2023.

[9] Mahitha Kumar. How to use drop() in pandas? *Numpyninja*, 2020.

[10] Michael Galarnyk. Understanding train test split. *Builtin*, 2022.

[11] Sebastian Taylor. Bayes' theorem.

[12] Saily Shah. Convolutional neural network: An overview. 2022.

[13] Afaque Umer. Understanding convolutional neural networks: A beginner's journey into the architecture. *Medium*, 2023.

[14] Mayank Mishra. Convolutional neural networks, explained. *Towards Data Science*, 2020.

[15] Harkiran78. Artificial neural networks and its applications. *geeksforgeeks*, 2023.

[16] Simeon Kostadinov. Understanding gru networks. *Towards Data Science*, 2017.

[17] Shipra Saxena. Introduction to the architecture of alexnet. *Analytics Vidhya*, 2023.