

HEART DISEASE PREDICTION USING CONVOLUTION NEURAL NETWORK

Submitted in partial fulfilment of the requirements for the award of
**Bachelor of Engineering Degree in
Electronics and Communication Engineering**

BY

JONNALA KARTHIK REDDY (38130090)

KANCHARLA NARAYANA KOUSHIK (38130099)



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING**

SCHOOL OF ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119, TAMILNADU

MARCH – 2022



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **JONNALA KARTHIK REDDY (38130101)**, **KANCHARLA NARAYAN KOUSHIK (38130099)** who carried out the project entitled "**HEART DISEASE PREDICTION USING CONVOLUTION NEURAL NETWORK**" under my supervision from June 2021 to November 2021.

Internal Guide

Dr. VIJAYABASKARAN, M.E., Ph.D.

Head of the Department

Dr. T. RAVI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **KANCHARLA NARAYAN KOUSHIK** hereby declare that the Project Report entitled “**PREDICTION OF HEART DISEASE USING CONVOLUTION NEURAL NETWORK**” done by me under the guidance of **Dr. VIJAYABASKARAN, M.E., Ph.D., (Internal)** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. N. NANDHITA, Ph.D.**, Dean, School of Computing **Dr. T RAVI M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. V. VIJAYABASKARAN, M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Electronics and communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The correct prediction of heart disease can prevent life threats, and incorrect prediction can prove to be fatal at the same time. In this paper different machine learning algorithms and deep learning are applied to compare the results and analysis of the UCI Machine Learning Heart Disease dataset. The dataset consists of 14 main attributes used for performing the analysis. Various promising results are achieved and are validated using accuracy and confusion matrix. The dataset consists of some irrelevant features which are handled using Isolation Forest, and data are also normalized for getting better results. And how this study can be combined with some multimedia technology like mobile devices is also discussed. Using deep learning approach, 94.2% accuracy was obtained.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xi
1	INTRODUCTION	1
	1.1 OBJECTIVE OF THE PROJECT	4
	1.2 LAYOUT OF THE DOCUMENT	4
	1.3 OVERVIEW OF THE DESIGNED PROJECT	4
	1.4 PROJECT PROPOSAL	4
	1.5 EXISTING SYSTEM	5
	1.6 PROPOSED SYSTEM	5
2	LITERATURE SURVEY	6
3	AIM AND SCOPE OF THE PROJECT	9
	3.1 AIM	9
	3.1.1 Required components	9
	3.1.1.1 Hardware required	9
	3.1.1.2 Software required	10
	3.2 PYTHON LIBRARIES NEEDED	10
	3.2.1 NumPy Library	10
	3.2.2 Pandas Library	10
	3.2.3 Matplotlib Library	10
	3.2.4 Seaborn Library	11
	3.2.5 Pathlib	11
	3.2.6 Scikit-Learn Library	11
	3.3 DATA PRE-PROCESSING	11
	3.4 DATA VALIDATION/ CLEANING / PREPARING PROCESS	13

CHAPTER NO	TITLE	PAGE NO
3.5	KERAS	13
3.6	WFDB	14
3.7	EXPLORATION DATA ANALYSIS OF VISUALIZATION	14
3.8	MACHINE LEARNING	15
3.8.1	How does Machine Learning work	16
3.8.2	Features of Machine Learning	17
3.8.3	Need for Machine Learning	17
3.8.4	Classification of Machine Learning	18
3.8.4.1	<i>Supervised Learning</i>	19
3.8.4.2	Unsupervised Learning	19
3.8.4.3	Reinforcement Learning	20
3.8.5	Logistic Regression in Machine Learning	20
3.9	RANDOM FOREST ALGORITHM	22
3.9.1	Decision Tree Classification Algorithm	23
3.9.2	Support Vector Machine Algorithm	24
3.9.2.1	<i>Types of SVM</i>	25
3.10	DEEP LEARNING	26
3.10.1	Architectures	27
3.10.2	<i>Types of Neural Network</i>	28
3.10.3	Convolution Neural Network	28
3.10.4	Auto Encoders	29
3.11	IMPORTANCE OF NEURAL NETWORK	29

CHAPTER NO	TITLE	PAGE NO
	3.12 LONG SHORT TERM MEMORY NETWORKS	30
	3.12.1 Tensor flow	30
	3.13 MICROCONTROLLER	35
	3.14 ARDUINO	38
	3.15 LABVIEW	40
	3.15.1 Features of LabVIEW	41
	3.15.2 Role of components	42
	3.15.3 Front panel	42
	3.15.4 Back Panel	43
4	RESULTS AND DISCUSSION AND PERFORMANCE ANALYSIS	45
	4.1 PERFORMANCE ANALYSIS	45
5	SUMMARY AND CONCLUSION	48
	5.1 SUMMARY	48
	5.2 CONCLUSION	48
	REFERENCE	49
	APPENDIX	50

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
3.1	Arduino programming	40

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.1	Block Diagram	9
3.2	Block Diagram Machine Learning algorithm	17
3.3	Classification of Machine Learning	18
3.4	Logistic Function	21
3.5	Random Forest algorithm	22
3.6	Decision Tree	24
3.7	Support Vector Machine	25
3.8	TensorFlow	32
3.9	Feature Columns	34
3.10	System bus connects	37
4.1	Confusion matrix for training data set	45
4.2	Confusion matrix for testing data set	46

LIST OF ABBREVIATIONS

CNN	-	Convolutional Neural Network
CPU	-	Central Processing Unit
DBN	-	Deep Belief Networks
DIP	-	Dual Inline Package
DNN	-	Deep Neural Network
ECG	-	Electrocardiogram
GPU	-	Graphical Processing Unit
LabVIEW	-	Laboratory Virtual Instrument Engineering Workbench
LEDs	-	Light Emitting Diodes
LSTM	-	Long Short Term Memory
ML	-	Machine Learning
RNN	-	Recurrent Neural Network
SVM	-	Support Vector Machine
WFDB	-	Waveform-Database

CHAPTER 1

INTRODUCTION

Heart disease describes a range of conditions that affect your heart. Today, cardiovascular diseases are the leading cause of death worldwide with 17.9 million deaths annually, as per the World Health Organization reports. Various unhealthy activities are the reason for the increase in the risk of heart disease like high cholesterol, obesity, increase in triglycerides levels, hypertension, etc. There are certain signs which the American Heart Association lists like the persons having sleep issues, a certain increase and decrease in heart rate (irregular heartbeat), swollen legs, and in some cases weight gain occurring quite fast; it can be 1-2 kg daily. All these symptoms resemble different diseases also like it occurs in the aging persons, so it becomes a difficult task to get a correct diagnosis, which results in fatality in near future.

But as time is passing, a lot of research data and patients records of hospitals are available. There are many open sources for accessing the patient's records and researches can be conducted so that various computer technologies could be used for doing the correct diagnosis of the patients and detect this disease to stop it from becoming fatal. Nowadays it is well known that machine learning and artificial intelligence are playing a huge role in the medical industry. We can use different machine learning and deep learning models to diagnose the disease and classify or predict the results. A complete genomic data analysis can easily be done using machine learning models. Models can be trained for knowledge pandemic predictions and also medical records can be transformed and analyzed more deeply for better predictions.

Many studies have been performed and various machine learning models are used for doing the classification and prediction for the diagnosis of heart disease. An automatic classifier for detecting congestive heart failure shows the patients at

high risk and the patients at low risk by Melillo et al. They used machine learning algorithm as CART which stands for Classification and Regression in which sensitivity is achieved as 93.3 percent and specificity is achieved as 63.5 percent. Then for improving the performance electrocardiogram (ECG) approach is suggested by Rahhal et al. In which deep neural networks are used for choosing the best features and then using them. Then, for detecting heart failures, a clinical decision support system is contributed by Guidi et al. for preventing it at an early stage. They tried to compare different machine learning models and deep learning models especially neural networks, as support vector machine, random forest, and CART algorithms. An 87.6 percent accuracy was achieved by random forest and CART, which outperformed everyone used in the classification. Combining the natural language processing with the rule-based approach, Zhang et al. achieved 93.37 percent accuracy when the NYHA HF class was found from the unstructured clinical notes. SVM techniques used for detecting patients who already have diabetes and then predicting heart disease by Parthiban and Srivatsa achieved a 94.60 percent accuracy rate, and the features taken were common like blood sugar level, age of the patient, and their blood pressure data.

In machine learning, a common problem is the high dimensionality of the data; the datasets which we use contain huge data and sometimes we cannot view that data even in 3D, which is also called the curse of dimensionality. So, when we perform operations on this data, we require a huge amount of memory, and sometimes the data can also grow exponentially and overfitting can happen. The weighting features can be used, so the redundancy in the dataset can be decreased which in turn also helps in decreasing the processing time of the execution. For decreasing the dimensionality of the dataset, there are various feature engineering and feature selection techniques which can be used to remove that data not having that much importance in the dataset.

In literature, when feature engineering and feature selection are applied, the results improve, both for classification as well as predictions, tried various machine learning and deep learning techniques for detecting the heart disease and also performed hyperparameters tuning for increasing the results accuracy. Neural networks achieved high accuracy of 78.3 percent, and the other models were logistic

regression, SVM, and ensemble techniques like Random Forest, etc. For reducing the cardiovascular features, Singh et al. used generalized discriminant analysis for extracting nonlinear features; a binary classifier like an extreme learning machine for less overfitting and increasing the training speed and the ranking method used for all these was Fisher. The accuracy achieved was 100 percent for detecting coronary heart disease. Arrhythmias classification was done by Yaghoubi et al. for heart rate variability. A multilayer perceptron neural network was used for doing the classification and 100 percent accuracy is achieved by reducing the features or Gaussian Discriminant Analysis, used Gaussian discriminant analysis for reducing the HRV signal features to 15 and 100 percent precision is achieved using the SVM classifier.

For dealing with data that are of high variance or high dimensional data, by using appropriate dimensionality reduction techniques like PCA, we can store valuable information in new components. PCA is used by many researchers as the first preference while dealing with high dimensionality data. Rajagopal and Ranganathan used five different dimensionality reduction techniques which are unsupervised (linear and nonlinear), and neural network is used as a classifier for classifying cardiac arrhythmia. FastICA (used for independent component analysis) with a minimum of 10 components was able to achieve an F1 score of 99.83 percent. Zhang et al. used the AdaBoost algorithm which is based on PCA for detecting breast cancer. Negi et al. combined uncorrelated discriminant analysis with PCA so that the best features that are used for controlling the upper limb motions can be selected and the results were great. Avendaño-Valencia et al. tried to reduce heart sounds to increase performance by applying PCA techniques on time-frequency representations. Kamencay. tried a new method for different medical images reaching an accuracy of 83.6 percent when trained on 200 images by using PCA-KNN which is a scale-invariant feature used in medical images for the scaling purpose. Ratnasari et al. used a gray-level threshold of 150 based on PCA and ROI, all of these used for reducing features of the X-ray images.

Heart disease is very fatal and it should not be taken lightly. Heart disease happens more in males than females, which can be read further from Harvard Health Publishing. Researchers found that, throughout life, men were about twice as likely

as women to have a heart attack. That higher risk persisted even after they accounted for traditional risk factors of heart disease, including high cholesterol, high blood pressure, diabetes, body mass index, and physical activity. The researchers are working on this dataset as it contains certain important parameters like dates from 1998, and it is considered as one of the benchmark datasets when someone is working on heart disease prediction. This dataset dates from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V, and the results achieved are quite promising.

1.1 OBJECTIVE OF THE PROJECT

The main goal of the project is to predict the heart disease of the people using ECG signal which further uses deep learning to predict the what kind of disease it was based on the past experience

1.2 LAYOUT OF THE DOCUMENT

This documentation starts with formal introduction. After introduction analysis and design of the project are described. In analysis and design of the project have many parts such as project proposal, mission, goal, target audience, environment. Use cases and test cases are in chapter 2 and chapter 3 respectively. Finally, this documentation finished with result and Conclusion part.

1.3 OVERVIEW OF THE DESIGNED PROJECT

At first in this project we will collect the data from the dataset then we cleaned the data and then we used the deep learning model to test the dataset after that we got the input directly through ECG sensor then we use that input and predict the output and that output is showed in labview.

1.4 PROJECT PROPOSAL

The project proposal is the term of documents. A project can describe the project proposal. It is the set of all plans of a project. Like, how the software works, what are the steps to complete the entire projects, and what are the software

requirements and analysis for this project. In my project, I am doing all the steps and also risk and reward and other project dependencies in the project proposal.

1.5 EXISTING SYSTEM

A variety of technologies and tests are currently available to evaluate underground cables but there is often little relation between the diagnostic results and the actual detracton. The failures of underground power distribution cables represents a serious threat to the reliability of power infrastructure. Replacement must be done selectively since cable replacement is expensive, being estimated at no less than hundred thousand dollars per kilometre of cable in area.

1.6 PROPOSED SYSTEM

- In this we are using neural networks to predict the disease accurately.
- We are going to predict that a person is having heart disease or not.
- ECG data is an uncertain dataset. Neural Network (NN) plays a major role in taking decision in these types of dataset. NN models are used to recognize ECG
- In our proposed system we can take both signal data and image data and we will predict the output

CHAPTER 2

LITERATURE SURVEY

Cardiologist-Level Arrhythmia Detection and Classification in Ambulatory Electrocardiograms Using a Deep Neural Network

Computerized electrocardiogram (ECG) interpretation plays a critical role in the clinical ECG workflow¹. Widely available digital ECG data and the algorithmic paradigm of deep learning² present an opportunity to substantially improve the accuracy and scalability of automated ECG analysis. However, a comprehensive evaluation of an end-to-end deep learning approach for ECG analysis across a wide variety of diagnostic classes has not been previously reported. Here, we develop a deep neural network (DNN) to classify 12 rhythm classes using 91,232 single-lead ECGs from 53,877 patients who used a single-lead ambulatory ECG monitoring device. When validated against an independent test dataset annotated by a consensus committee of board-certified practicing cardiologists, the DNN achieved an average area under the receiver operating characteristic curve (AUC) of 0.97. The average F1 score, which is the harmonic mean of the positive predictive value and sensitivity, for the DNN (0.837) exceeded that of average cardiologists (0.780). With specificity fixed at the average specificity achieved by cardiologists, the sensitivity of the DNN exceeded the average cardiologist sensitivity for all rhythm classes. These findings demonstrate that an end-to-end deep learning approach can classify a broad range of distinct arrhythmias from single-lead ECGs with high diagnostic performance similar to that of cardiologists. If confirmed in clinical settings, this approach could reduce the rate of misdiagnosed computerized ECG interpretations and improve the efficiency of expert-human ECG interpretation by accurately triaging or prioritizing the most urgent conditions.

Classification of Arrhythmia in Heartbeat Detection Using Deep Learning

The electrocardiogram (ECG) is one of the most widely used diagnostic instruments in medicine and healthcare. Deep learning methods have shown promise in healthcare prediction challenges involving ECG data. This paper aims to apply deep learning techniques on the publicly available dataset to classify arrhythmia. We have used two kinds of the dataset in our research paper. One dataset is the MIT-BIH arrhythmia database, with a sampling frequency of 125 Hz with 1,09,446 ECG beats. The classes included in this first dataset are N, S, V, F, and Q. The second database is PTB Diagnostic ECG Database. The second database has two classes. The techniques used in these two datasets are the CNN model, CNN + LSTM, and CNN + LSTM + Attention Model. 80% of the data is used for the training, and the remaining 20% is used for testing. The result achieved by using these three techniques shows the accuracy of 99.12% for the CNN model, 99.3% for CNN + LSTM, and 99.29% for CNN + LSTM + Attention Model.

ECG Signal Detection and Classification of Heart Rhythm Diseases Based on ResNet and LSTM

In this paper, based on the ResNet34 network, a three-layer stacked long-term and short-term memory networks are added, and the Mish function is used as the activation function. The final improved model can obtain the context dependence of the feature and retain the negative information in the ECG signal. The average F1 score of 0.861 and the average AUC value of 0.972 are obtained by the improved ResNet34-LSTM3 model on the PhysioNet challenge 2017 test dataset, which shows that the model can effectively extract the characteristics of ECG signals and diagnose arrhythmia diseases. Comparing the evaluation results of the previous ResNet34 model and ResNet18 model on the same test dataset, it can be seen that the improved model has a better classification and recognition effect on ECG signals as a whole, and it can more effectively identify arrhythmias such as atrial fibrillation, which will provide a more effective and reliable diagnostic basis for medical workers. There are some important limitations in this study. The input dataset of the experiment is PhysioNet challenge 2017 Short Single-Lead ECG signal, which provides a limited signal compared with the standard 12 lead ECG

signal. Therefore, whether the ResNet34-LSTM3 model classification performance is better in the 12 lead ECG signals remains to be determined. In addition, when the algorithm is used clinically, it may be limited by the duration of ECG signals, and the application of all kinds of algorithms, including the one presented algorithm, must eventually tailor specific ECG signal pretreatment methods for the target clinical application. Therefore, in the next stage of the study, we consider segmenting the signal to supplement the signal segment by copying other electrocardiogram signals in the same category in order to maximize the use of information. In the future, we will conduct experiments with more types of ECG data to prove the performance of our model. In a word, the ResNet34-LSTM3 network model in this paper can distinguish the signals with different concentric laws in Short Single-Lead ECG signals, and its classification performance is also better than that of the predecessors in partial scores. If more tests are carried out in the clinical environment, this method may help medical workers improve the efficiency and accuracy of ECG clinical interpretation.

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

3.1 AIM

The main aim of the project is to predict the heart disease using the deep learning model which is used to predict the output of the electrocardiogram graph which is given as the input to the deep learning model and the output is shown in the Labview.

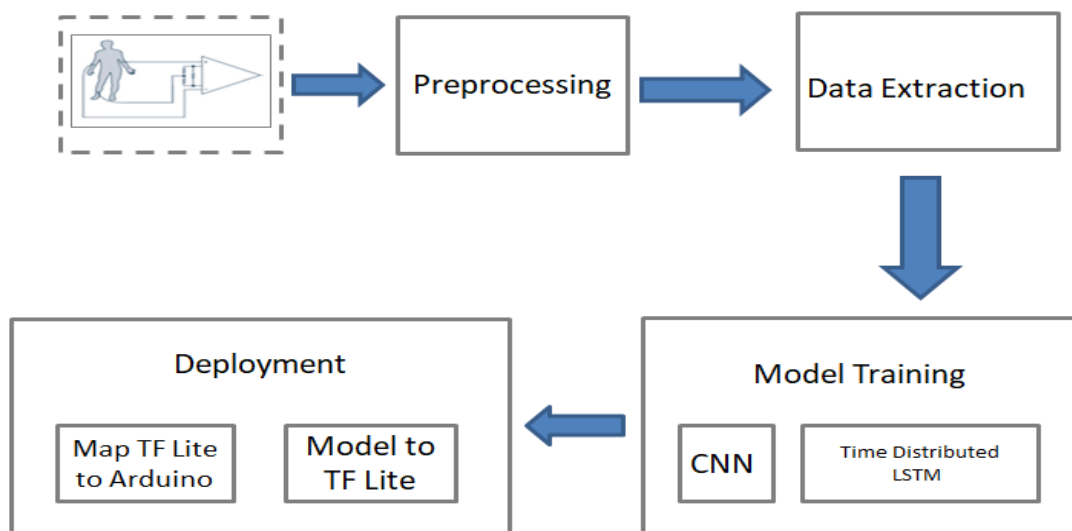


Fig. 3.1: Block Diagram

3.1.1 Required components

3.1.1.1 Hardware required

Ecg monitor sensor

Arduino nano

3.1.1.2 *Software required*

Arduino IDE

Tensorflow

Labview

Labview vision module

Python libraries

Deep learning(lstm model)

3.2 **PYTHON LIBRARIES NEEDED**

3.2.1 ***NumPy Library***

NumPy is an open-source numerical Python library. NumPy contains a multi- dimensional array and matrix data structures. It can be utilized to perform a number of mathematical operations on arrays such as trigonometric, statistical, and algebraic routines like mean, mode, standard deviation etc...,

3.2.2 ***Pandas Library***

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables. There are several ways to create a DataFrame.

3.2.3 ***Matplotlib Library***

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Use interactive figures that can zoom, pan, update, visualize etc...,

Here we use pyplot mainly for plotting graphs.

matplotlib.pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc

3.2.4 *Seaborn Library*

Seaborn package was developed based on the Matplotlib library. It is used to create more attractive and informative statistical graphics. While seaborn is a different package, it can also be used to develop the attractiveness of matplotlib graphics.

3.2.5 *Pathlib*

The Pathlib is a python library which provides an object api for working with files and directories

3.2.6 *Scikit-Learn Library*

Scikit-learn is a free machine learning library for the Python. It features various algorithms like support vector machine, random forests, regression and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

3.3 DATA PRE-PROCESSING

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers use this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model.

A number of different data cleaning tasks using Python Pandas library and specifically, it focus on probably the biggest data cleaning task, missing values and it able to more quickly clean data. It wants to spend less time cleaning data, and more time exploring and modeling.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It's important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it's important to understand the sources of missing data. Here are some typical reasons why data is missing.

Variable identification with Uni-variate, Bi-variate and Multi-variate analysis:

- import libraries for access and functional purpose & read the given dataset
- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- show columns
- shape of the data frame
- To describe the data frame

- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

3.4 DATA VALIDATION/ CLEANING/PREPARING PROCESS

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process.

The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

3.5 KERAS

Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

However, Keras is also a highly-flexible framework suitable to iterate on state-of-the-art research ideas. Keras follows the principle of **progressive disclosure of complexity**: it makes it easy to get started, yet it makes it possible to handle arbitrarily advanced use cases, only requiring incremental learning at each step.

In much the same way that you were able to train & evaluate a simple neural network above in a few lines, you can use Keras to quickly develop new training procedures or exotic model architectures. Here's a low-level training loop example, combining Keras functionality with the TensorFlow

If you need to, you can further configure your optimizer. The Keras philosophy is to keep simple things simple, while allowing the user to be fully in control when they need to (the ultimate control being the easy extensibility of the source code via subclassing)

3.6 WFDB

The waveform-database (WFDB) package for Python is a library of tools for reading, writing, and processing physiological signals and annotations, written in the Python programming language. Core components of the Python package are loosely based on the specifications of the original C-language WFDB software

3.7 EXPLORATION DATA ANALYSIS OF VISUALIZATION

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more.

With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

3.8 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning

algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies.

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

3.8.1 *How does Machine Learning work*

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it.** The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking

about the problem. The below block diagram explains the working of Machine Learning algorithm:

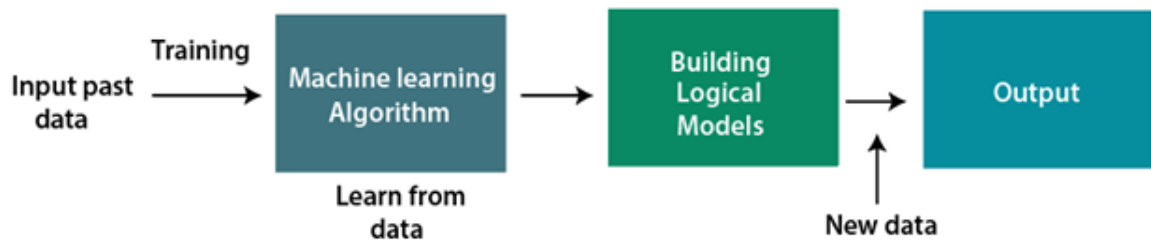


Fig. 3.2: Block Diagram Machine Learning algorithm

3.8.2 Features of Machine Learning

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

3.8.3 Need for Machine Learning

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm

depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in **self-driving cars**, **cyber fraud detection**, **face recognition**, and **friend suggestion by Facebook**, etc. Various top companies such as Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly

3.8.4 *Classification of Machine Learning*

At a broad level, machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**
3. **Reinforcement learning**

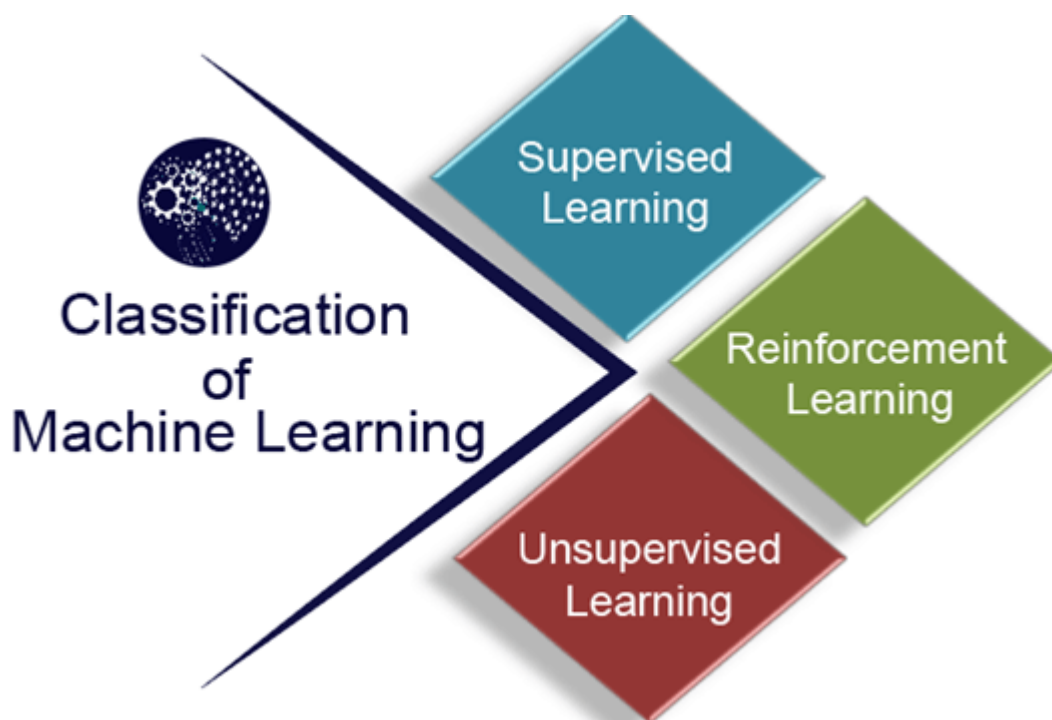


Fig. 3.3: Classification of Machine Learning

3.8.4.1 *Supervised Learning*

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**.

Supervised learning can be grouped further in two categories of algorithms:

- **Classification**
- **Regression**

3.8.4.2 *Unsupervised Learning*

Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classified into two categories of algorithms:

- **Clustering**
- **Association**

3.8.4.3 *Reinforcement Learning*

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

These 5 different algorithms are compared:

- Logistic Regression
- Random Forest
- Decision Tree
- Support Vector Machine
- K Nearest Neighbor

3.8.5 *Logistic Regression in Machine Learning*

- 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.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

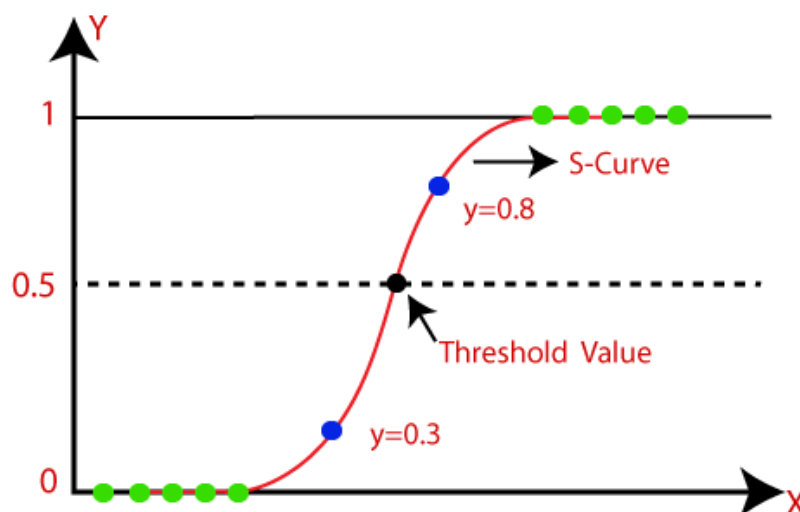


Fig. 3.4: Logistic Function

3.9 RANDOM FOREST ALGORITHM

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning**, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model*.

As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:

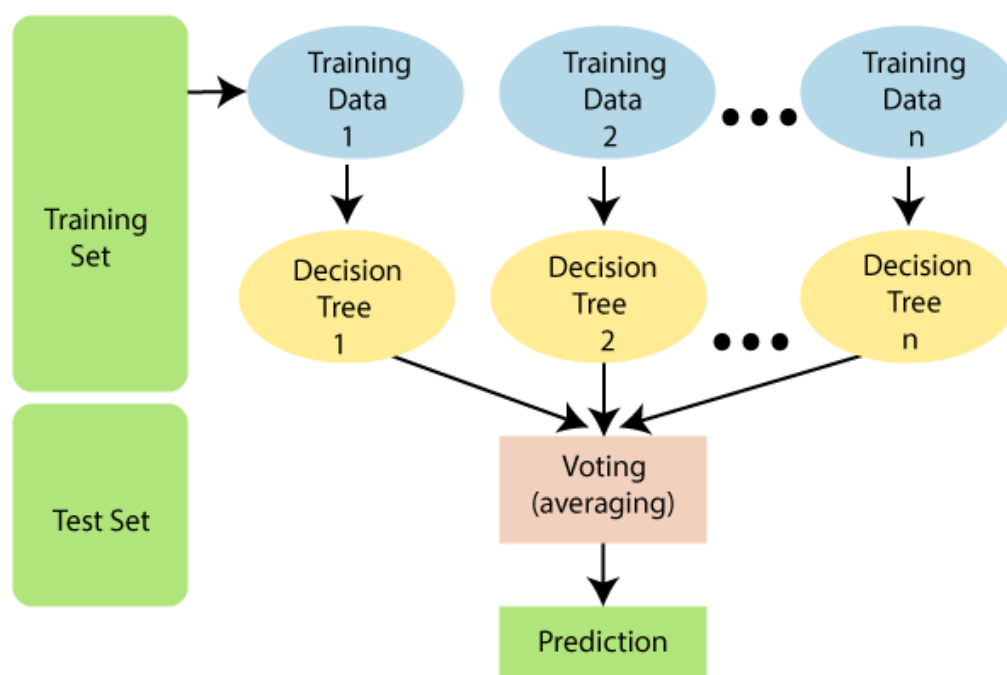


Fig. 3.5: Random Forest algorithm

3.9.1 *Decision Tree Classification Algorithm*

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome**.
- In a Decision tree, there are two nodes, which are the **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.
- The decisions or the test are performed on the basis of features of the given dataset.
- ***It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.***
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm**, which stands for **Classification and Regression Tree algorithm**.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:

Note: A decision tree can contain categorical data (YES/NO) as well as numeric data

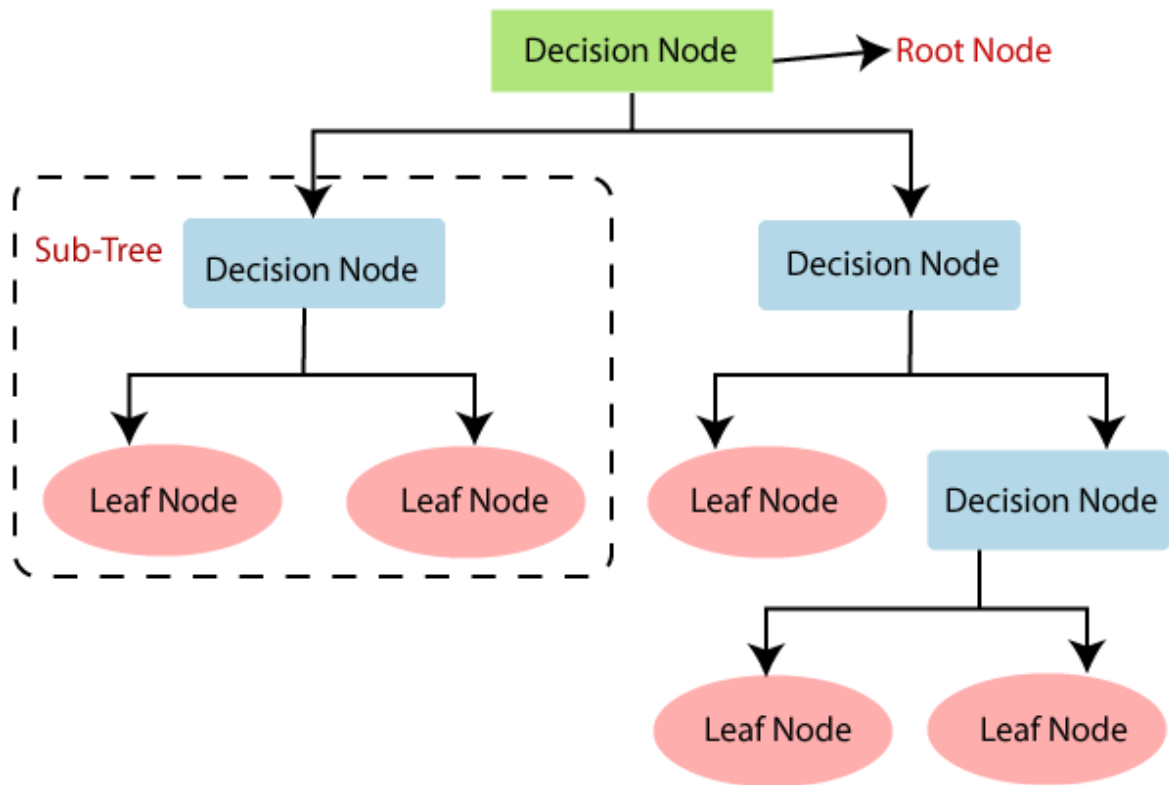


Fig. 3.6: Decision Tree

3.9.2 Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

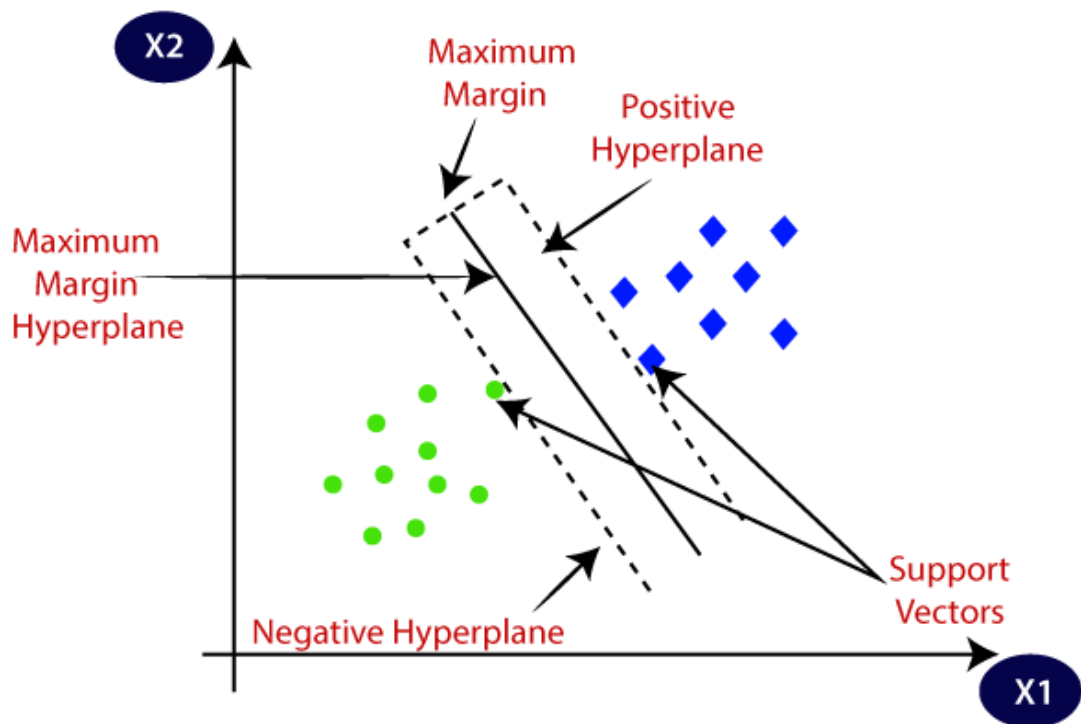


Fig. 3.7: Support Vector Machine

3.9.2.1 Types of SVM

SVM can be of two types

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best

decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

3.10 DEEP LEARNING

Deep learning is based on the branch of machine learning, which is a subset of artificial intelligence. Since neural networks imitate the human brain and so deep learning will do. In deep learning, nothing is programmed explicitly. Basically, it is a machine learning class that makes use of numerous nonlinear processing units so as to perform feature extraction as well as transformation. The output from each preceding layer is taken as input by each one of the successive layers.

Deep learning models are capable enough to focus on the accurate features themselves by requiring a little guidance from the programmer and are very helpful in solving out the problem of dimensionality. Deep learning algorithms are used, especially when we have a huge no of inputs and outputs.

Since deep learning has been evolved by the machine learning, which itself is a subset of artificial intelligence and as the idea behind the artificial intelligence is to mimic the human behavior, so same is "the idea of deep learning to build such algorithm that can mimic the brain".

Deep learning is implemented with the help of Neural Networks, and the idea behind the motivation of Neural Network is the biological neurons, which is nothing but a brain cell.

3.10.1 Architectures

Deep Neural Networks

It is a neural network that incorporates the complexity of a certain level, which means several numbers of hidden layers are encompassed in between the input and output layers. They are highly proficient on model and process non-linear associations.

Deep Belief Networks

A deep belief network is a class of Deep Neural Network that comprises of multi-layer belief networks.

1. With the help of the Contrastive Divergence algorithm, a layer of features is learned from perceptible units.
2. Next, the formerly trained features are treated as visible units, which perform learning of features.
3. Lastly, when the learning of the final hidden layer is accomplished, then the whole DBN is trained.

Recurrent Neural Networks

It permits parallel as well as sequential computation, and it is exactly similar to that of the human brain (large feedback network of connected neurons). Since they are capable enough to reminisce all of the imperative things related to the input they have received, so they are more precise.

3.10.2 *Types of Neural Network*

Feed Forward Neural Network

A feed-forward neural network is none other than an Artificial Neural Network, which ensures that the nodes do not form a cycle. In this kind of neural network, all the perceptrons are organized within layers, such that the input layer takes the input, and the output layer generates the output. Since the hidden layers do not link with the outside world, it is named as hidden layers. Each of the perceptrons contained in one single layer is associated with each node in the subsequent layer. It can be concluded that all of the nodes are fully connected. It does not contain any visible or invisible connection between the nodes in the same layer. There are no back-loops in the feed-forward network. To minimize the prediction error, the backpropagation algorithm can be used to update the weight values.

Recurrent Neural Network

Recurrent neural network are yet another variation of feed-forward networks. Here each of the neurons present in the hidden layers receives an input with a specific delay in time. The Recurrent neural network mainly accesses the preceding info of existing iterations. For example, to guess the succeeding word in any sentence, one must have knowledge about the words that were previously used. It not only processes the inputs but also shares the length as well as weights crossways time. It does not let the size of the model to increase with the increase in the input size. However, the only problem with this recurrent neural network is that it has slow computational speed as well as it does not contemplate any future input for the current state. It has a problem with reminiscing prior information.

3.10.3 *Convolution Neural Network*

Convolution neural network are a special kind of neural network mainly used for image classification, clustering of images and object recognition. DNNs enable unsupervised construction of hierarchical image representations. To achieve

the best accuracy, deep convolutional neural networks are preferred more than any other neural network

Restricted Boltzmann machine

rbms are a special kind of neural network mainly used for image classification, clustering of images and object recognition. DNNs enable unsupervised construction of hierarchical image representations. To achieve the best accuracy, deep convolutional neural networks are preferred more than any other neural network.

3.10.4 Auto Encoders

An autoencoder neural network is another kind of unsupervised machine learning algorithm. Here the number of hidden cells is merely small than that of the input cells. But the number of input cells is equivalent to the number of output cells. An autoencoder network is trained to display the output similar to the fed input to force AEs to find common patterns and generalize the data. The autoencoders are mainly used for the smaller representation of the input. It helps in the reconstruction of the original data from compressed data. This algorithm is comparatively simple as it only necessitates the output identical to the input.

3.11 IMPORTANCE OF NEURAL NETWORK

Deep learning algorithms play a crucial role in determining the features and can handle the large number of processes for the data that might be structured or unstructured. Although, deep learning algorithms can overkill some tasks that might involve complex problems because they need access to huge amounts of data so that they can function effectively. For example, there's a popular deep learning tool that recognizes images namely **Imagenet** that has access to **14 million** images in its dataset-driven algorithms. It is a highly comprehensive tool that has defined a next-level benchmark for deep learning tools that aim images as their dataset.

Deep learning algorithms are highly progressive algorithms that learn about the image that we discussed previously by passing it through each neural

network layer. The layers are highly sensitive to detect low-level features of the image like **edges** and **pixels** and henceforth the combined layers take this information and form holistic representations by comparing it with previous data. For example, the middle layer might be programmed to detect some special parts of the object in the photograph which other deep trained layers are programmed to detect special objects like **dogs, trees, utensils**, etc.

3.12 LONG SHORT TERM MEMORY NETWORKS

LSTM can be defined as **Recurrent Neural Networks** (RNN) that are programmed to learn and adapt for dependencies for the long term. It can memorize and recall past data for a greater period and by default, it is its sole behavior. LSTMs are designed to retain over time and henceforth they are majorly used in time series predictions because they can restrain memory or previous inputs. This analogy comes from their **chain-like** structure consisting of **four** interacting layers that communicate with each other differently. Besides applications of time series prediction, they can be used to construct **speech recognizers, development in pharmaceuticals**, and composition of **music loops** as well.

LSTM work in a sequence of events. First, they don't tend to remember irrelevant details attained in the previous state. Next, they update certain cell-state values selectively and finally generate certain parts of the cell-state as output. Below is the diagram of their operation.

3.12.1 *Tensor flow*

TensorFlow tutorial is designed for both beginners and professionals. Our tutorial provides all the basic and advanced concept of machine learning and deep learning concept such as deep neural network, image processing and sentiment analysis.

TensorFlow is one of the famous deep learning framework, developed by **Google** Team. It is a free and open source software library and designed in **Python** programming language, this tutorial is designed in such a way that we

can easily implement deep learning project on TensorFlow in an easy and efficient way.

ensorFlow is a popular framework of **machine learning** and **deep learning**. It is a **free** and **open-source** library which is released on **9 November 2015** and developed by **Google Brain Team**. It is entirely based on Python programming language and use for numerical computation and data flow, which makes machine learning faster and easier.

TensorFlow can train and run the deep neural networks for image recognition, handwritten digit classification, recurrent neural network, **word embedding**, **natural language processing**, video detection, and many more. TensorFlow is run on multiple **CPUs** or **GPUs** and also mobile operating systems.

The name TensorFlow is derived from its core framework, "**Tensor**." A tensor is a vector or a matrix of n-dimensional that represents all type of data. All values in a tensor hold similar data type with a known shape. The shape of the data is the dimension of the matrix or an array.

A tensor can be generated from the input data or the result of a computation. In TensorFlow, all operations are conducted inside a graph. The group is a set of calculation that takes place successively. Each transaction is called an op node are connected.

TensorFlow makes use of a graph framework. The chart gathers and describes all the computations done during the training.

TensorFlow is the better library for all because it is accessible to everyone. TensorFlow library integrates different **API** to create a scale deep learning architecture like **CNN (Convolutional Neural Network)** or **RNN (Recurrent Neural Network)**.

TensorFlow is based on graph computation; it can allow the developer to create the construction of the neural network with Tensorboard. This tool helps

debug our program. It runs on CPU (Central Processing Unit) and GPU (Graphical Processing Unit).

ensorFlow has an interactive **multiplatform** programming interface which is scalable and reliable compared to other deep learning libraries which are available.

These features of TensorFlow will tell us about the popularity of TensorFlow.

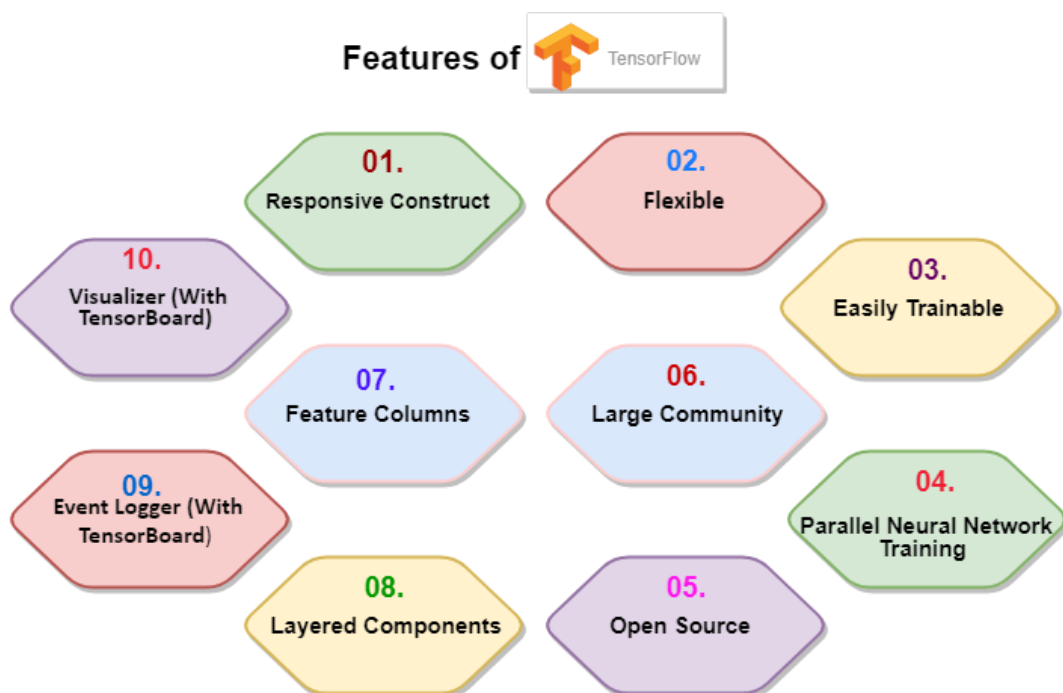


Fig. 3.8: TensorFlow

Responsive Construct

We can visualize each part of the graph, which is not an option while using **Numpy** or **SciKit**. To develop a deep learning application, firstly, there are two or three components that are required to create a deep learning application and need a programming language.

Flexible

It is one of the essential TensorFlow Features according to its operability. It has modularity and parts of it which we want to make standalone.

Easily Trainable

It is easily trainable on CPU and for GPU in distributed computing.

Parallel Neural Network Training

TensorFlow offers to the pipeline in the sense that we can train multiple neural networks and various **GPUs**, which makes the models very efficient on large-scale systems.

Large Community

Google has developed it, and there already is a large team of software engineers who work on stability improvements continuously.

Open Source

The best thing about the machine learning library is that it is open source so anyone can use it as much as they have internet connectivity. So, people can manipulate the library and come up with a fantastic variety of useful products. And it has become another **DIY** community which has a massive forum for people getting started with it and those who find it hard to use it.

Feature Columns

TensorFlow has feature columns which could be thought of as intermediates between raw data and estimators; accordingly, **bridging** input data with our model.

The feature below describes how the feature column is implemented.

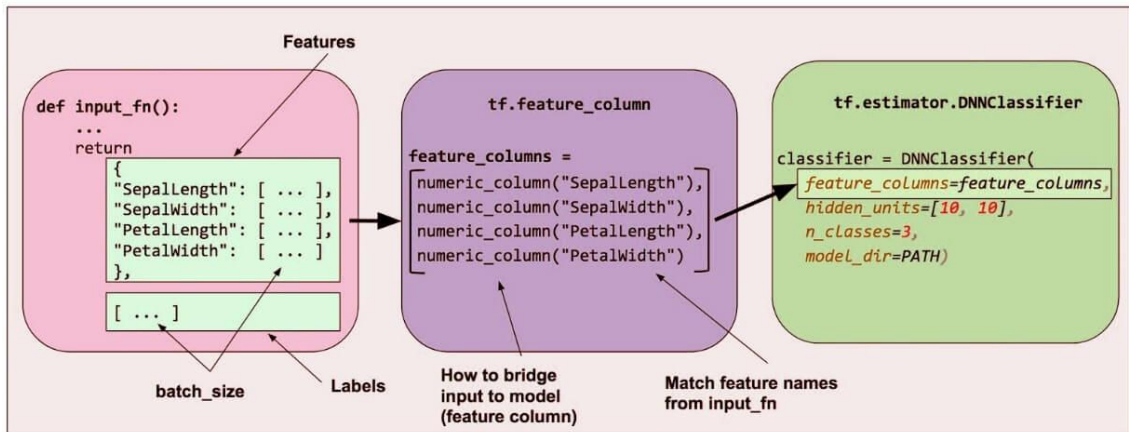


Fig. 3.9: Feature Columns

Availability of Statistical Distributions

This library provides distributions functions including Bernoulli, Beta, Chi2, Uniform, Gamma, which are essential, especially where considering probabilistic approaches such as Bayesian models.

Layered Components

TensorFlow produces layered operations of weight and biases from the function such as `tf.contrib.layers` and also provides batch normalization, convolution layer, and dropout layer. So **`tf.contrib.layers.optimizers`** have optimizers such as **Adagrad**, **SGD**, **Momentum** which are often used to solve optimization problems for numerical analysis.

Visualizer (With TensorBoard)

We can inspect a different representation of a model and make the changed necessary while debugging it with the help of TensorBoard.

Event Logger (With TensorBoard)

It is just like UNIX, where we use **tail - f** to monitor the output of tasks at the cmd. It checks, logging events and summaries from the graph and production with the TensorBoard.

3.13 MICROCONTROLLER

A **microcontroller** is a small and low-cost microcomputer, which is designed to perform the specific tasks of embedded systems like displaying microwave's information, receiving remote signals, etc.

The general microcontroller consists of the processor, the memory (RAM, ROM, EPROM), Serial ports, peripherals (timers, counters), etc.

Microcontrollers are divided into various categories based on memory, architecture, bits and instruction sets. Following is the list of their types –

Bit

Based on bit configuration, the microcontroller is further divided into three categories.

- **8-bit microcontroller** – This type of microcontroller is used to execute arithmetic and logical operations like addition, subtraction, multiplication division, etc. For example, Intel 8031 and 8051 are 8 bits microcontroller.
- **16-bit microcontroller** – This type of microcontroller is used to perform arithmetic and logical operations where higher accuracy and performance is required. For example, Intel 8096 is a 16-bit microcontroller.
- **32-bit microcontroller** – This type of microcontroller is generally used in automatically controlled appliances like automatic operational machines, medical appliances, etc.

Memory

Based on the memory configuration, the microcontroller is further divided into two categories.

- **External memory microcontroller** – This type of microcontroller is designed in such a way that they do not have a program memory on the chip. Hence, it is named as external memory microcontroller. For example: Intel 8031 microcontroller.
- **Embedded memory microcontroller** – This type of microcontroller is designed in such a way that the microcontroller has all programs and data memory, counters and timers, interrupts, I/O ports are embedded on the chip. For example: Intel 8051 microcontroller.

Instruction Set

Based on the instruction set configuration, the microcontroller is further divided into two categories.

- **CISC** – CISC stands for complex instruction set computer. It allows the user to insert a single instruction as an alternative to many simple instructions.
- **RISC** – RISC stands for Reduced Instruction Set Computers. It reduces the operational time by shortening the clock cycle per instruction.

Applications of Microcontrollers

Microcontrollers are widely used in various different devices such as –

- Light sensing and controlling devices like LED.
- Temperature sensing and controlling devices like microwave oven, chimneys.
- Fire detection and safety devices like Fire alarm.
- Measuring devices like Volt Meter.

- 8051 microcontroller is designed by Intel in 1981. It is an 8-bit microcontroller. It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, 2 16-bit timers. It consists of are four parallel 8-bit ports, which are programmable as well as addressable as per the requirement. An on-chip crystal oscillator is integrated in the microcontroller having crystal frequency of 12 MHz.
- Let us now discuss the architecture of 8051 Microcontroller.
- In the following diagram, the system bus connects all the support devices to the CPU. The system bus consists of an 8-bit data bus, a 16-bit address bus and bus control signals. All other devices like program memory, ports, data memory, serial interface, interrupt control, timers, and the CPU are all interfaced together through the system bus.

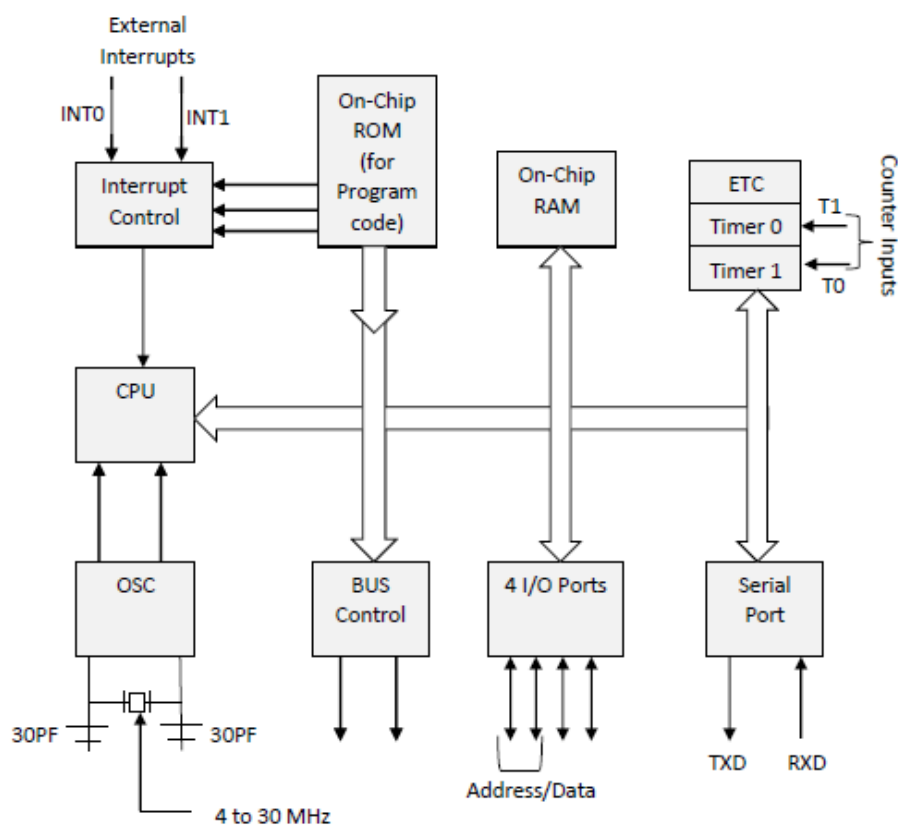


Fig. 3.10: System bus connects

3.14 ARDUINO

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are –

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.
- Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programmed through the Arduino IDE.
- The differences are based on the number of inputs and outputs (the number of sensors, LEDs, and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface

(hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need at least 5V.

In this chapter, we will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below, but most Arduinos have majority of these components in common.

After learning about the main parts of the Arduino UNO board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1 – First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

Step 2 – Download Arduino IDE Software - You can get different versions of Arduino IDE from the [Arduino Official website](#). You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

Step 3 – Power up your board - The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to your computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 – Launch Arduino IDE - After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.

Data types in C refers to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

The following table provides all the data types that you will use during Arduino programming.

Table 3.1: Arduino programming

void	Boolean	char	Unsigned char	byte	int	Unsigned int	word
long	Unsigned long	short	float	double	array	String-char array	String-object

3.15 LABVIEW

Graphical Programming Technique is a technique where **VISUAL BLOCK** Connections are used to code instead of text which makes it easy for non-coders to implement algorithms. **LabVIEW(Laboratory Virtual Instrument Engineering Workbench)** is the first implementation of graphical programming to date, it continues to be the dominant graphical programming implementation. It provides a **powerful and integrated environment** for the development of various instrumental applications. An efficient LabVIEW application is designed without unnecessary operations, with minimal occupation including code, data, block diagram, front panel, and GUI updates. It **eliminates human errors** in data

collection and process operations. It **reduces data transcription errors** and more reliable data available makes better quality control of products and new discoveries. LabVIEW programs are also called **virtual instruments** (VIs), because their appearance and operation imitate physical instruments. It contains a comprehensive set of VIs and functions for acquiring, analyzing, displaying, and storing data, as well as tools to help you troubleshoot your code. It also has built-in features for connecting user's applications to the Web using the **LabVIEW Web Server**. It is used to manage large and professional applications and also integrated project management tools, integrated graphical debugging tools, and standardized source code control integration. LabVIEW provides the tools required for most applications and is also an open development environment.

3.15.1 *Features of LabVIEW*

Some other features of Graphical Programming are:

- **User-friendly UI:** It has a user-friendly drag and drop kind of interactive User Interface.
- **Built-in Functions:** It supports thousands of inbuilt functions that range from analysis and I/O etc. These belong to the function palette.
- **Scalable:** As LabVIEW has a modular design making it easy to scale and modular programs.
- **Professional Development Tools:** It has a plethora of tools that help integrate and debug large applications.
- **Open environment:** It has tools needed for many open environment developments.
- **Object-oriented design:** It supports object-oriented programming structures enabling encapsulation and inheritance to create modular and extensible code.
- **Compiled language:** Being a compiled language it is faster.

3.15.2 *Role of components*

- A LabVIEW constitutes a graphical programming environment which allows user to design and analyze any complicated system in a shorter time as compared with the text-based programming environment.
- The LabVIEW graphical programs are called **virtual instruments**.
- When data is available at all the inputs, the block or the graphical component is executed.
- After the execution is completed, the data is supplied to output terminals, and then it is passed to the next block in the dataflow path.

3.15.3 *Front panel*

The front panel allows the user to interact with the VI by displaying outputs and allowing the user to supply the program with inputs.

Main front panel objects are:

Controls and Indicators

- The controls simulate the input devices and supply the data to the block diagram of the VI. The common controls are knobs, pushbuttons, dials, and other input devices.
- The indicators Simulate output devices and display data that is acquired or generated by the block diagram. The common indicators are graphs, Light Emitting Diodes (LEDs), meters, and other output devices.

3.15.4 Back Panel

- The back panel consists of the code used by the VI to obtain inputs from the front panel, operate on the inputs, and display the results.
- The back panel is also known as the block diagram.
- Block diagram controls the program through graphical code.
- In the block diagram, the code is added using a graphical representation of functions that control the front panel objects.
- The back panel contains structures and functions that perform operations on controls and supply data to indicators.

The three different **palettes** available in LabVIEW are :

Tool Palette

- A tool is a special operating mode of the mouse cursor. The cursor corresponds to the icon of the tool selected in the Tools palette.
- Using Tool Palette users can create, modify and debug Virtual Instruments.
- The tool palette is available on both the front panel and the Block Diagram.

Different tools available are as follows:

- **Operating tool button:** For text selection or for changing the values of controls.
- **Positioning Tool:** For resizing, selecting, and positioning.
- **Labelling Tool:** For free labels and editing text.
- **Object shortcut Menu Tool:** For opening the shortcut menu of an object.

Controls Palette

- The Controls Palette is available only on the Front panel.
- It consists of various controls and indicators which are required by the user while building the front panel.

Function Palette

It is available only on the Block Diagram and is used to build Block Diagram.

Different function palettes include :

- Numeric
- Array
- Time and Dialog
- Waveform

LabVIEW has built-in libraries for integrating stand-alone instruments, data acquisition devices, motion control, and vision products. For instance, take a look at the below image depicting Data Acquisition (DAQ)

CHAPTER 4

RESULTS AND DISCUSSION AND PERFORMANCE ANALYSIS

4.1 PERFORMANCE ANALYSIS

The model what we have used for training data set has the accuracy of 98.46 percentage and for testing set we got 90.31 percentage accuracy now we have drawn the confusion matrix for the testing dataset and training dataset which was shown below and the time taken for the model from giving input to till the output is 2 milliseconds

For this type of model prediction we must not only consider the accuracy we must also consider precision and recall.

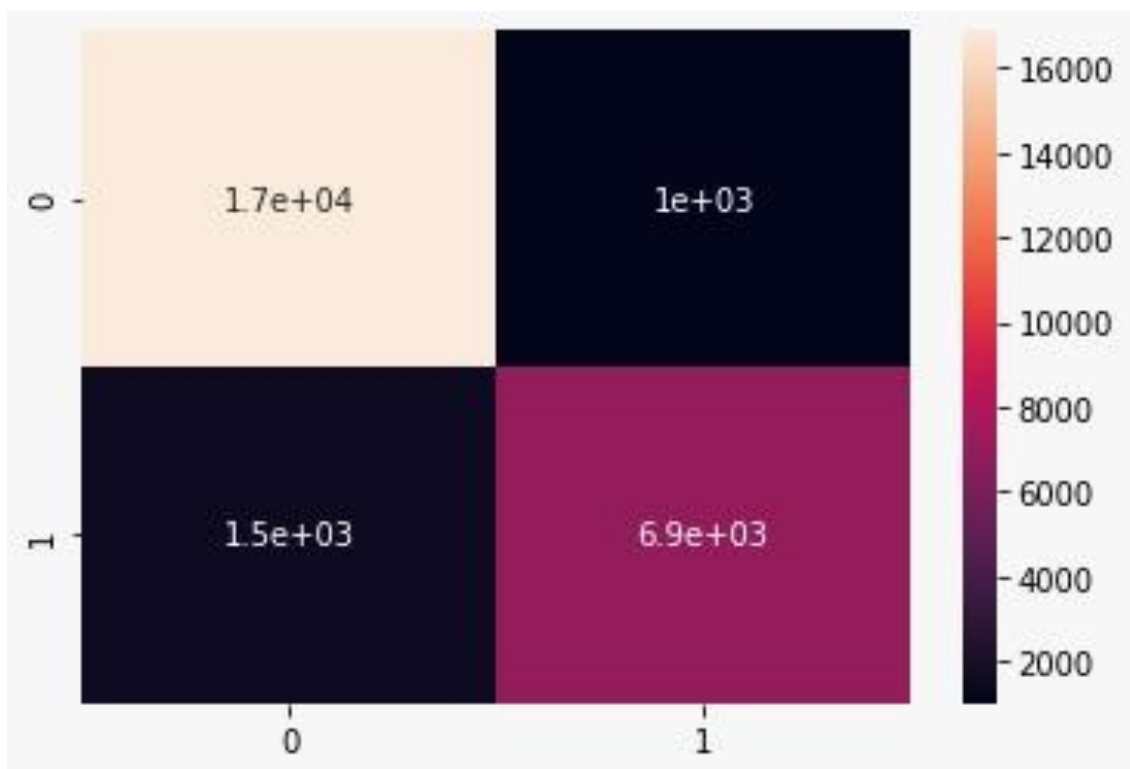


Fig. 4.1: Confusion matrix for training data set

This matrix is for testing dataset

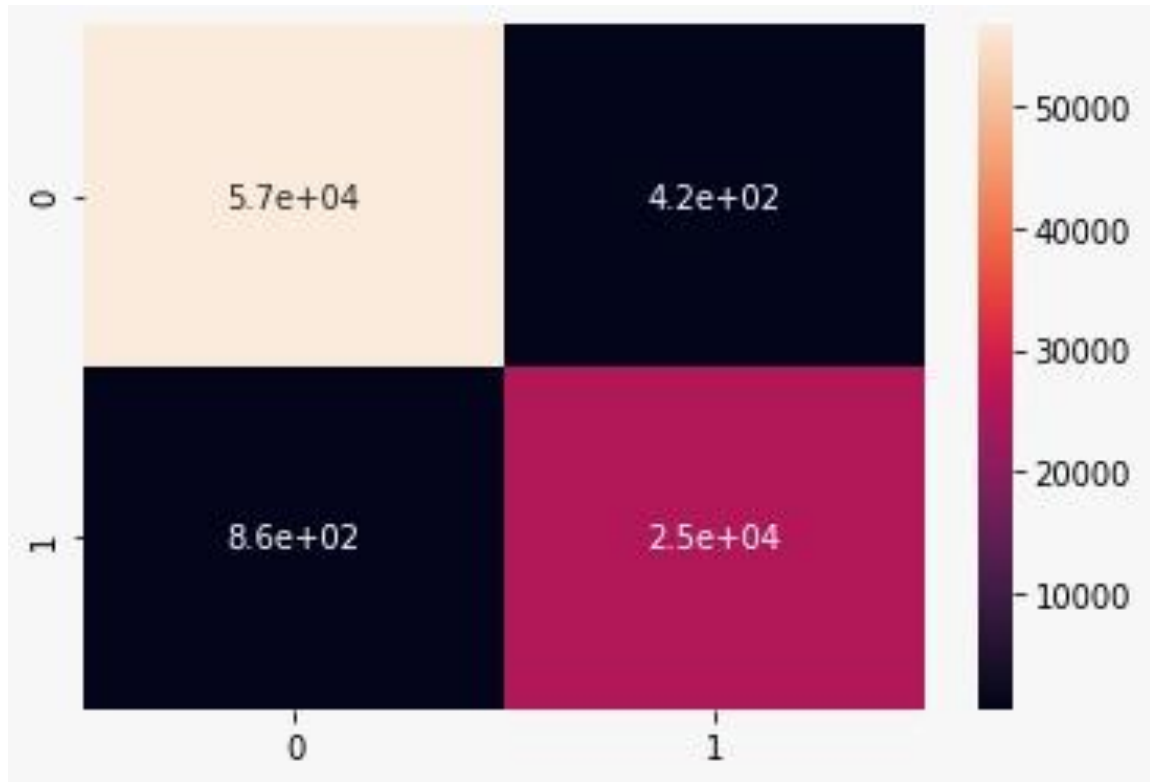


Fig. 4.2: Confusion matrix for testing data set

This matrix is for training data set

Precision and recall score calculation for training set

$$\begin{aligned}\text{Precision} &= \text{TP} / \text{TP} + \text{FP} \\ &= 56704 / 56704 + 421 \\ &= 0.98351 \\ \text{Recall} &= \text{TP} / \text{TP} + \text{FN} \\ &= 56704 / 56704 + 858 \\ &= 0.96697\end{aligned}$$

Precision and recall score calculation for testing set

$$\begin{aligned}\text{Precision} &= \text{TP} / \text{TP} + \text{FP} \\ &= 16867 / 16867 + 1019 \\ &= 0.87155 \\ \text{Recall} &= \text{TP} / \text{TP} + \text{FN} \\ &= 16867 / 16867 + 1533 \\ &= 0.81852\end{aligned}$$

We got better recall and precision for both the training and testing data set.

CHAPTER 5

SUMMARY AND CONCLUSION

5.1 SUMMARY

The main idea of the project is to predict the heart disease so to help the people which disease they are having for the people so that people can get that before getting to the hospital

5.2 CONCLUSION

The process started from the cleaning and training the model and then we are predicting the output and then we are displaying the output in the labview.

REFERENCE

- [1] MITArthmiyyadatabase(<https://physionet.org/content/mitdb/1.0.0/>)
- [2] wfdb (<https://wfdb.readthedocs.io/en/latest/>)
- [3] Tensor flow(<https://www.tensorflow.org>)
- [4] Tensor flow lite(<https://www.tensorflow.org/lite>)

APPENDIX

A) Screenshots

