

A Major-Project report on
**LIVER CIRRHOSIS PREDICTION USING MACHINE LEARNING
ALGORITHMS**

Submitted in partial fulfillment of the
requirements for the award of the degree

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING
SUBMITTED BY

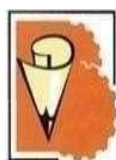
M.HARISHWAR REDDY	(18P61A04D7)
P.PAVAN KUMAR	(18P61A04G8)
R.SURYAPRAKASH REDDY	(17P61A04E8)

Under the Esteemed Guidance of

Mrs.T.PUSHPA

Associate Professor

Department of Electronics and Communication Engineering



VIGNANA BHARATHI
Institute of Technology



(Affiliated to JNTU Hyderabad, Approved by APSCHE & AICTE) Aushapur (v), Ghatkesar
(m), Medchal Dist, Hyderabad-501301
2021-2022



VIGNANA BHARATHI
Institute of Technology



Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the Major Project report **“LIVER CIRRHOSIS PREDICTION USING MACHINE LEARNING ALGORITHMS”** being submitted by **M.Harishwar Reddy(18P61A04D7), P.Pavan Kumar(18P61A04G8),R.Surya Prakash Reddy(17P61A04E8)** in partial fulfillment for the award of the Degree Bachelor of Technology in **ELECTRONICS AND COMMUNICATION ENGINEERING** to Jawaharlal Nehru Technological University is a record of a bonafide work carried out by them under my guidance and supervision

The result embodied in this project report has not been submitted to any other University/Institution for the award of any Degree/Diploma.

Internal Guide

Mrs.T.Pushpa

Assistant Professor

Head of the Department

Dr.U.Poorna Lakshmi

Professor

CANDIDATES DECLARATION

We hereby declare that this Major Project Report titled **“LIVER CIRRHOSIS PREDICTION USING MACHINE LEARNING ALGORITHMS ”** submitted by us to the Department of **ELECTRONICS AND COMMUNICATION ENGINEERING**, VBIT, Aushapur, Under JNTUH, is a Bonafide work undertaken by and it is not submitted to any other University or Institution for the award of any degree or diploma.

By:

M.HARISHWAR REDDY (18P61A04D7)

P.PAVAN KUMAR (18P61A04G8)

R.SURYAPRAKASH REDDY (17P61A04E8)

ACKNOWLEDGEMENT

At the outset we sincerely thank God for having got my Major project report completed in time. Firstly, we would thank our parents who have been a motivating factor throughout our lives. Secondly, we sincerely thank our principal **Dr. PVS Srinivas** and our Head of the department **Dr. U.Poorna Lakshmi** for their kind cooperation and Encouragement for the successful completion of Seminar work and providing the necessary facilities.

We are most obliged and grateful to our project guide **Mrs T.Pushpa**, for giving us guidance in completing this project successfully.

We express our sincere gratitude to our Project coordinators, Department of ECE and my other faculty for attending our project seminars and for their insightful comments and constructive suggestions to improve the quality of this project work.

By

M.HARISHWAR REDDY (18P61A04D7)

P.PAVAN KUMAR (18P61A04G8)

R.SURYAPRAKASH REDDY (17P61A04E8)

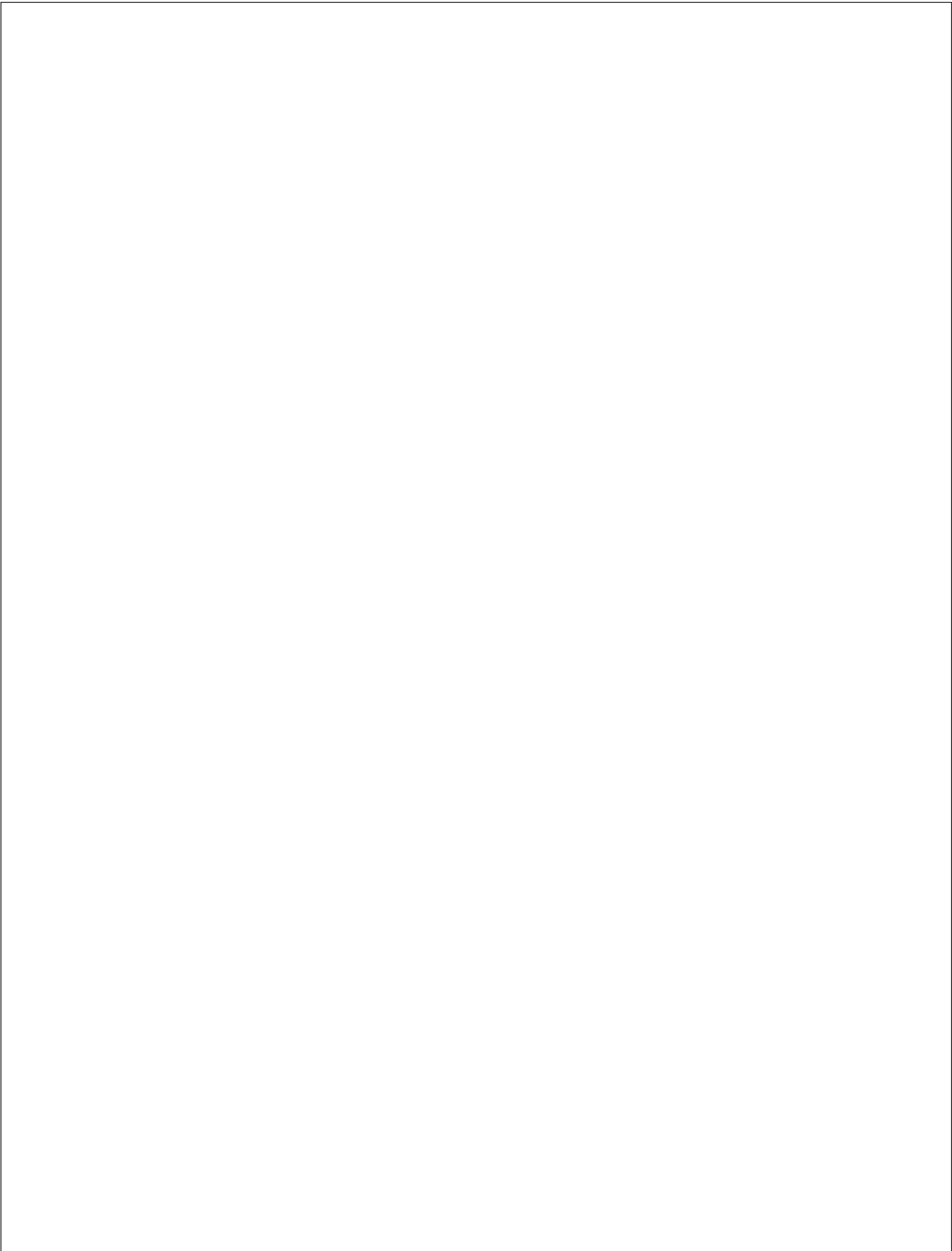
ABSTRACT

With a growing trend of sedentary and lack of physical activities, diseases related to liver have become a common encounter nowadays. In rural areas the intensity is still manageable, but in urban areas, and especially metropolitan areas the liver disease is a very common sighting nowadays. Liver diseases cause millions of deaths every year. NASH has a number of causes, including obesity, high blood pressure, abnormal levels of cholesterol, type 2 diabetes, and metabolic syndrome. Less common causes of cirrhosis include autoimmune hepatitis, primary biliary cholangitis, and primary sclerosing cholangitis that disrupts bile duct function, genetic disorders such as Wilson's disease and hereditary hemochromatosis, and chronic heart failure with liver congestion. In this paper, the risk of liver disease was predicted using two different machine learning algorithms. The final output was predicted based on the most accurate machine learning algorithm. Based on the accuracy we designed a system which asks a person to enter the details of his/her medical report. Then the system uses the most accurate model which is trained to predict, whether a person has risk of liver disease or not.

TABLE OF CONTENTS

CANDIDATES DECLARATION	I
ACKNOWLEDGEMENT	I
CHAPTER 1: INTRODUCTION	1
1.1 OVERVIEW OF LIVER AND LIVER CIRRHOSIS	1
CHAPTER 2: LITERATURE SURVEY	3
CHAPTER 3: EXISTING MODEL	4
3.1 LOGISTIC FUNCTION (SIGMOID FUNCTION):	5
3.2 STEPS IN LOGISTIC REGRESSION	5
CHAPTER 4: PROPOSED MODEL	7
4.1 SYSTEM OPTIMIZATION:	8
4.2 SYSTEM REQUIREMENTS	8
CHAPTER 5 : MACHINE LEARNING	9
5.1 SUPERVISED LEARNING:	9
5.2 UNSUPERVISED LEARNING:	9
5.3 SEMI-SUPERVISED LEARNING:	10
5.4 REINFORCEMENT LEARNING:	10
CHAPTER-6: SOFTWARE ENVIRONMENT	12
6.1 WHAT IS PYTHON	12
6.2 MODULES USED IN PROJECT	15
6.3 INSTALL PYTHON STEP-BY-STEP IN WINDOWS	16
6.4 INSTALLATION OF PYTHON	18

CHAPTER 7: IMPLEMENTATION	22
7.1 METHODOLOGY PROPOSED	22
7.2 HOW TO BUILD AN INTUITION FOR XGBOOST?	23
7.3 ALGORITHMIC ENHANCEMENTS:	24
7.4 ADVANTAGES OF XGBOOST ALGORITHM	24
7.5 WORKING OF XGBOOST	25
7.6 MATHEMATICS BEHIND XGBOOST	28
7.7 FLOWCHART OF OUR PROJECT	30
CHAPTER 8: GRAPHICAL USER INTERFACE IMPLEMENTATION	31
8.1 TKINTER PROGRAMMING	31
CHAPTER 9: RESULTS	35
9.1 EXISTING MODEL ACCURACY	35
9.2 PROPOSED MODEL ACCURACY	36
9.3 GRAPHICAL USER INTERFACE	37



CHAPTER 1: INTRODUCTION

Machine Learning techniques nowadays have become very much important in the healthcare sector for the prediction of disease from the medical database. Many researchers and companies are leveraging machine learning to improve medical diagnostics. Among different machine learning techniques, classification algorithms are widely used in predicting diseases. In this paper, Logistic Regression, Extreme gradient boosting algorithms are been used for prediction of liver disease. We all know that liver is the body largest internal organ which performs very important body function including making blood clotting factors and proteins, manufacturing triglyceride and cholesterol, glycogen synthesis and bile production. Usually, more than 75% of liver tissue needs to be affected by a decrease in function to occur. So it's important to detect at an early stage such that the disease can be treated before it becomes severe.

1.1 OVERVIEW OF LIVER AND LIVER CIRRHOSIS

The liver is a major organ only found in vertebrates which performs many essential biological functions such as detoxification of the organism, and the synthesis of proteins and biochemicals necessary for digestion and growth. In humans, it is located in the right upper quadrant of the abdomen, below the diaphragm. Its other roles in metabolism include the regulation of glycogen storage, decomposition of red blood cells, and the production of hormones.

The liver is an accessory digestive organ that produces bile, an alkaline fluid containing cholesterol and bile acids, which helps the breakdown of fat. The gallbladder, a small pouch that sits just under the liver, stores bile produced by the liver which is afterwards moved to the small intestine to complete digestion. The liver's highly specialized tissue, consisting of mostly hepatocytes, regulates a wide variety of high-volume biochemical reactions, including the synthesis and breakdown of small and complex molecules, many of which are necessary for normal vital functions.

Cirrhosis, also known as liver cirrhosis or hepatic cirrhosis, and end-stage liver disease, is the impaired liver function caused by the formation of scar tissue known as fibrosis due to damage caused by liver disease. Damage causes tissue repair and subsequent formation of scar tissue, which over time can replace normal functioning tissue, leading to the impaired liver function of cirrhosis. The disease typically develops slowly over months or years. Early symptoms may include tiredness, weakness, loss of appetite, unexplained weight loss, nausea and vomiting, and discomfort in the right upper quadrant of the abdomen. As the disease worsens, symptoms may include itchiness, swelling in the lower legs, fluid build-up in the abdomen, jaundice, bruising easily, and the development of spider-like blood vessels in the skin. The fluid

build-up in the abdomen may become spontaneously infected. More serious complications include hepatic encephalopathy, bleeding from dilated veins in the esophagus, stomach, or intestines, and liver cancer.



Figure 1.1 Difference between healthy and cirrhosis liver

Cirrhosis is most commonly caused by alcoholic liver disease, non-alcoholic steatohepatitis (NASH – the progressive form of non-alcoholic fatty liver disease), chronic hepatitis B, and chronic hepatitis C. Heavy drinking over a number of years can cause alcoholic liver disease. NASH has a number of causes, including obesity, high blood pressure, abnormal levels of cholesterol, type 2 diabetes, and metabolic syndrome. Less common causes of cirrhosis include autoimmune hepatitis, primary biliary cholangitis, and primary sclerosing cholangitis that disrupts bile duct function, genetic disorders such as Wilson's disease and hereditary hemochromatosis, and chronic heart failure with liver congestion.

Common causes are listed below as follows:

- Alcoholic liver disease (ALD, or alcoholic cirrhosis) develops for 10–20% of individuals who drink heavily for a decade or more. Alcohol seems to injure the liver by blocking the normal metabolism of protein, fats, and carbohydrates. This injury happens through the formation of acetaldehyde from alcohol. Acetaldehyde is reactive and leads to the accumulation of other reactive products in the liver. People with ALD may also have concurrent alcoholic hepatitis. Associated symptoms are fever, hepatomegaly, jaundice, and anorexia. AST and ALT blood levels are both elevated, but at less than 300 IU/liter, with an AST:ALT ratio > 2.0 , a value rarely seen in other liver diseases. In the United States, 40% of cirrhosis-related deaths are due to alcohol.
- In non-alcoholic fatty liver disease (NAFLD), fat builds up in the liver and eventually causes scar tissue. This type of disorder can be caused by obesity, diabetes, malnutrition, coronary artery disease, and steroids. Though similar in signs to alcoholic liver disease, no history of notable alcohol use is found. Blood tests and medical imaging are used to diagnose NAFLD and NASH, and sometimes a liver biopsy is needed.

CHAPTER 2: LITERATURE SURVEY

PAPER 1: Published in: 2018 4th International Conference on Computing Communication and Automation (ICCCA)

The proposed methods used are to compare classification accuracy of Logistic Regression, K-nearest neighbour and Support Vector Machine. The first step is to clean the data. Filling the missing values followed by transforming nominal attribute to binary attribute. The next step is feature selection to select the best attribute for a subset of features. Based on the findings, attributes been selected. The third step is data transformation. In this technique, data is been standardized such that the data follows Gaussian Distribution with a mean of 0 and standard deviation of 1. In the fourth step, the classification model is trained to predict the results in unseen data. In the last step, based on the accuracy of different classification model the best model is selected for the prediction of liver disease.

PAPER 2: Published in: 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT).

In this they have compared several ML methods, such as Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Extra Trees (ET) for the prediction of liver disorders. At the preprocessing step, categorical values are encoded through label encoding. After the completion of data preprocessing, LR, DT, RF, and ET classifiers are used for classifying liver disorders patients. For further improvement, the AdaBoost classifier is used to increase the performance of each classification algorithm.

CHAPTER 3: EXISTING MODEL

In the existing model we use logistic regression to predict the liver cirrhosis. This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables.

- 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:

On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
- **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

- **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

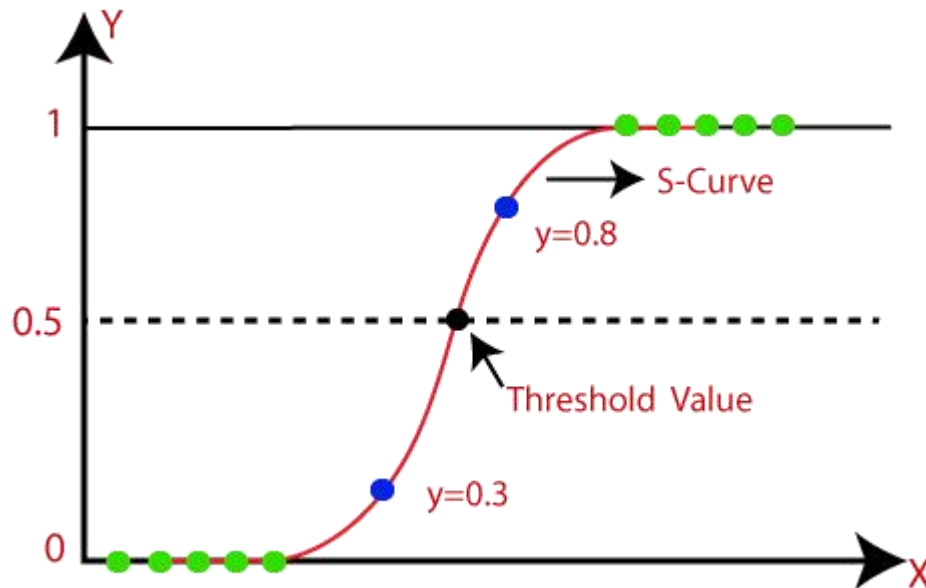


Figure 3.1 Logistic function

3.1 LOGISTIC FUNCTION (SIGMOID FUNCTION):

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Assumptions for Logistic Regression:

- The dependent variable must be categorical in nature.
- The independent variable should not have multi-collinearity.

3.2 STEPS IN LOGISTIC REGRESSION

To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result(Creation of Confusion matrix)
- Visualizing the test set result.

CHAPTER 4: PROPOSED MODEL

In the proposed model we use xgboost algorithm to predict. XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now. Please see the chart below for the evolution of tree-based algorithms over the years.

XGBoost algorithm was developed as a research project at the University of Washington. Tianqi Chen and Carlos Guestrin presented their paper at SIGKDD Conference in 2016 and caught the Machine Learning world by fire. Since its introduction, this algorithm has not only been credited with winning numerous Kaggle competitions but also for being the driving force under the hood for several cutting-edge industry applications. As a result, there is a strong community of data scientists contributing to the XGBoost open source projects

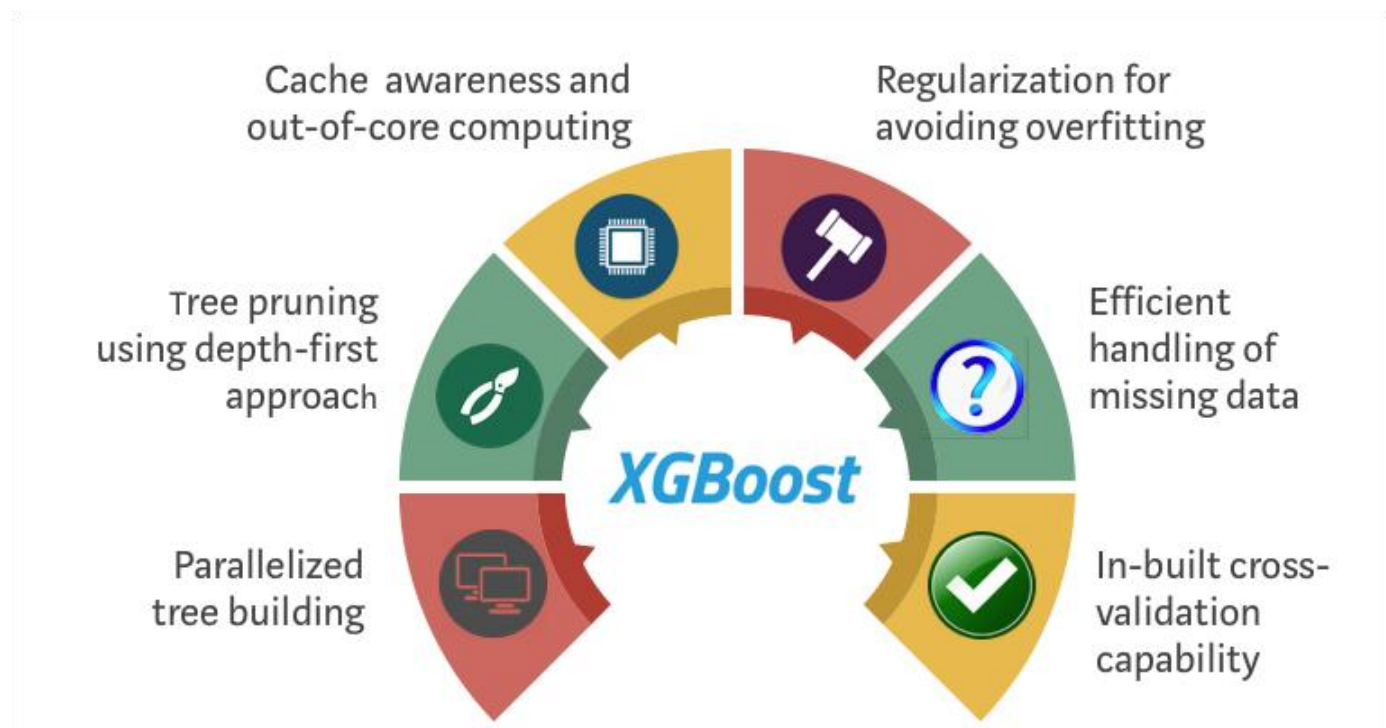


Figure 4.1 How XGBoost optimizes standard GBM algorithm

XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods that apply the principle of boosting weak learners (CARTs generally) using the gradient descent architecture. However, XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements.

4.1 SYSTEM OPTIMIZATION:

1. **Parallelization:** XGBoost approaches the process of sequential tree building using parallelized implementation. This is possible due to the interchangeable nature of loops used for building base learners; the outer loop that enumerates the leaf nodes of a tree, and the second inner loop that calculates the features. This nesting of loops limits parallelization because without completing the inner loop (more computationally demanding of the two), the outer loop cannot be started. Therefore, to improve run time, the order of loops is interchanged using initialization through a global scan of all instances and sorting using parallel threads. This switch improves algorithmic performance by offsetting any parallelization overheads in computation.
2. **Tree Pruning:** The stopping criterion for tree splitting within GBM framework is greedy in nature and depends on the negative loss criterion at the point of split. XGBoost uses ‘max_depth’ parameter as specified instead of criterion first, and starts pruning trees backward. This ‘depth-first’ approach improves computational performance significantly.
3. **Hardware Optimization:** This algorithm has been designed to make efficient use of hardware resources. This is accomplished by cache awareness by allocating internal buffers in each thread to store gradient statistics. Further enhancements such as ‘out-of-core’ computing optimize available disk space while handling big data-frames that do not fit into memory.

4.2 SYSTEM REQUIREMENTS

4.2.1 Hardware Requirements:

- CPU: Intel Core i5 9th gen
- GPU: Nvidia GTX 1650 4GB graphics card
- RAM: 8GB
- HDD: 120GB SSD
- System Type: 64-bit Operating System

4.2.2 Software Requirements

- Operating System: Windows
- Coding Language: Python 3.7
- Code Editor: Visual studio code, Jupyter notebook

CHAPTER 5 : MACHINE LEARNING

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

There are four basic approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.

5.1 SUPERVISED LEARNING:

In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

Supervised machine learning requires the data scientist to train the algorithm with both labeled inputs and desired outputs. Supervised learning algorithms are good for the following tasks:

- **Binary classification:** Dividing data into two categories.
- **Multi-class classification:** Choosing between more than two types of answers.
- **Regression modeling:** Predicting continuous values.
- **Ensembling:** Combining the predictions of multiple machine learning models to produce an accurate prediction.
-

5.2 UNSUPERVISED LEARNING:

This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

Unsupervised machine learning algorithms do not require data to be labeled. They sift through unlabeled data to look for patterns that can be used to group data points into subsets. Most types of deep learning,

including neural networks, are unsupervised algorithms. Unsupervised learning algorithms are good for the following tasks:

- **Clustering:** Splitting the dataset into groups based on similarity.
- **Anomaly detection:** Identifying unusual data points in a data set.
- **Association mining:** Identifying sets of items in a data set that frequently occur together.
- **Dimensionality reduction:** Reducing the number of variables in a data set.

5.3 SEMI-SUPERVISED LEARNING:

This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.

Semi-supervised learning works by data scientists feeding a small amount of labeled training data to an algorithm. From this, the algorithm learns the dimensions of the data set, which it can then apply to new, unlabeled data. The performance of algorithms typically improves when they train on labeled data sets. But labeling data can be time consuming and expensive. Semi-supervised learning strikes a middle ground between the performance of supervised learning and the efficiency of unsupervised learning. Some areas where semi-supervised learning is used include:

- **Machine translation:** Teaching algorithms to translate language based on less than a full dictionary of words.
- **Fraud detection:** Identifying cases of fraud when you only have a few positive examples.
- **Labelling data:** Algorithms trained on small data sets can learn to apply data labels to larger sets automatically.

5.4 REINFORCEMENT LEARNING:

Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

Reinforcement learning works by programming an algorithm with a distinct goal and a prescribed set of rules for accomplishing that goal. Data scientists also program the algorithm to seek positive rewards -- which it receives when it performs an action that is beneficial toward the ultimate goal -- and avoid punishments -- which it receives when it performs an action that gets it farther away from its ultimate goal. Reinforcement learning is often used in areas such as:

- **Robotics:** Robots can learn to perform tasks the physical world using this technique.
- **Video gameplay:** Reinforcement learning has been used to teach bots to play a number of video games.
- **Resource management:** Given finite resources and a defined goal, reinforcement learning can help enterprises plan out how to allocate resources

CHAPTER-6: SOFTWARE ENVIRONMENT

6.1 WHAT IS PYTHON

Below are some facts about Python.

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, instagram, Dropbox, Uber... etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

6.1.1 ADVANTAGES OF PYTHON

1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading,databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

2. Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

4. Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

6. Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

8. Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

9. Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run

it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

6.1.2 ADVANTAGES OF PYTHON OVER OTHER LANGUAGES :

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and deep learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

6.1.3 DISADVANTAGES OF PYTHON

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the client side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbon NELLE. The reason it is not so famous despite the existence of Bryton is that it isn't that secure.

3. Design Restrictions

As you know, Python is dynamically-typed. This means that you don't need to declare the type of variable while writing the code. It uses duck-typing. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

4. Underdeveloped Database Access Layers

Compared to more widely used technologies like JDBC (Java Database Connectivity) and ODBC (Open Database Connectivity), Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

6.2 MODULES USED IN PROJECT

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

6.3 INSTALL PYTHON STEP-BY-STEP IN WINDOWS

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace. The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows. There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner

who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheat sheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www.python.org>



Figure 6.1: Official site

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Figure 6.2: Download tab

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web Based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

6.4 INSTALLATION OF PYTHON

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.

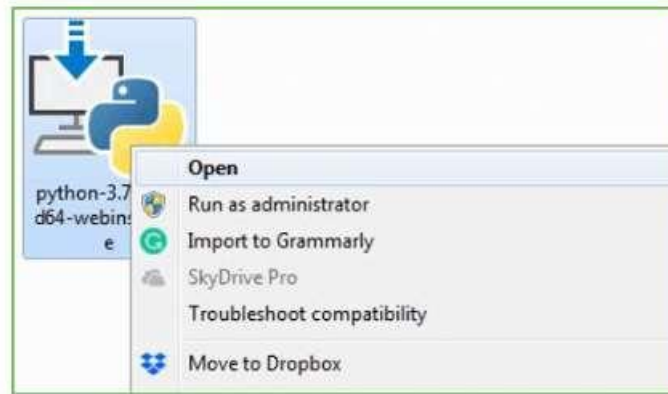


Figure 6.3: Python version

Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



Figure 6.4: Adding python path

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Figure 6.5: Installation

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes. Verify the Python Installation

Step 1: Click on Start

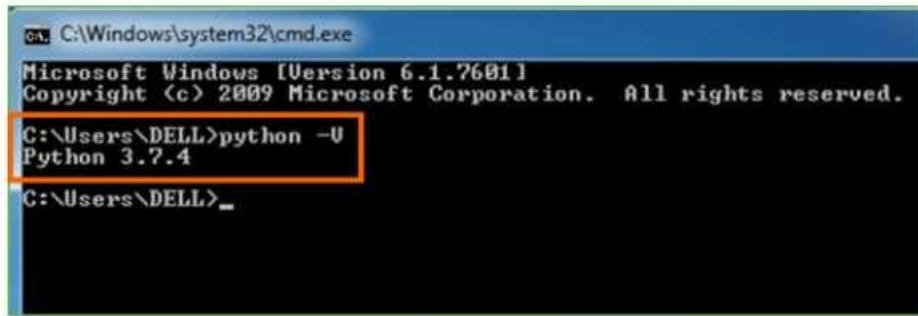
Step 2: In the Windows Run Command, type “cmd”.



Figure 6.6: Open command prompt

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python -V** and press Enter.



```
CA: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\DELL>python -U
Python 3.7.4
C:\Users\DELL>_
```

Figure 6.7: Version details in command prompt

We will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

CHAPTER 7: IMPLEMENTATION

7.1 METHODOLOGY PROPOSED

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now. Please see the chart below for the evolution of tree-based algorithms over the years.

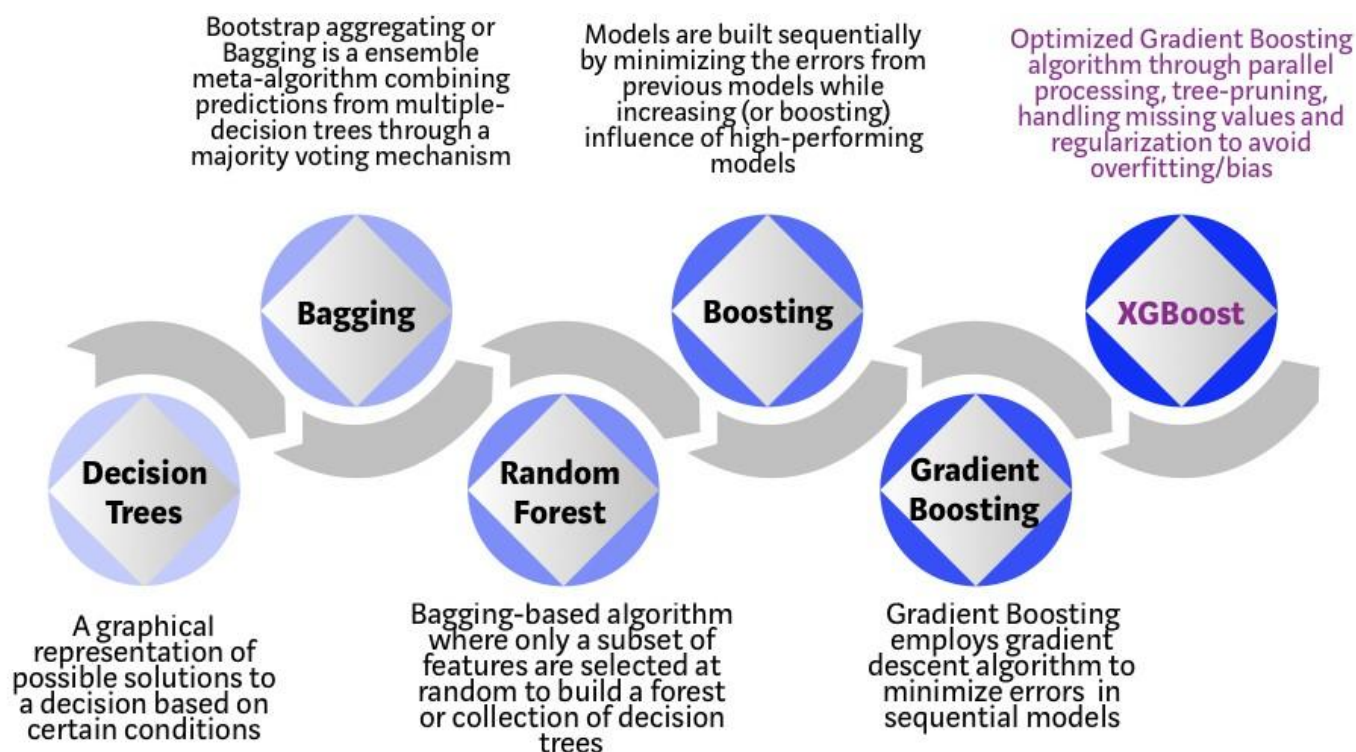


Figure 7.1 :Evolution of XGBoost Algorithm from Decision Trees

XGBoost algorithm was developed as a research project at the University of Washington. Tianqi Chen and Carlos Guestrin presented their paper at SIGKDD Conference in 2016 and caught the Machine Learning world by fire. Since its introduction, this algorithm has not only been credited with winning numerous Kaggle competitions but also for being the driving force under the hood for several cutting-edge industry applications. As a result, there is a strong community of data scientists contributing to the XGBoost open source projects with ~350 contributors and ~3,600 commits on GitHub. The algorithm differentiates itself in the following ways:

1. A wide range of applications: Can be used to solve regression, classification, ranking, and user-defined prediction problems.

2. **Portability:** Runs smoothly on Windows, Linux, and OS X.
3. **Languages:** Supports all major programming languages including C++, Python, R, Java, Scala, and Julia.
4. **Cloud Integration:** Supports AWS, Azure, and Yarn clusters and works well with Flink, Spark, and other ecosystems.

7.2 HOW TO BUILD AN INTUITION FOR XGBOOST?

Decision trees, in their simplest form, are easy-to-visualize and fairly interpretable algorithms but building intuition for the next-generation of tree-based algorithms can be a bit tricky. See below for a simple analogy to better understand the evolution of tree-based algorithms. Imagine that you are a hiring manager interviewing several candidates with excellent qualifications. Each step of the evolution of tree-based algorithms can be viewed as a version of the interview process.

1. **Decision Tree:** Every hiring manager has a set of criteria such as education level, number of years of experience, interview performance. A decision tree is analogous to a hiring manager interviewing candidates based on his or her own criteria.
2. **Bagging:** Now imagine instead of a single interviewer, now there is an interview panel where each interviewer has a vote. Bagging or bootstrap aggregating involves combining inputs from all interviewers for the final decision through a democratic voting process.
3. **Random Forest:** It is a bagging-based algorithm with a key difference wherein only a subset of features is selected at random. In other words, every interviewer will only test the interviewee on certain randomly selected qualifications (e.g. a technical interview for testing programming skills and a behavioral interview for evaluating non-technical skills).
4. **Boosting:** This is an alternative approach where each interviewer alters the evaluation criteria based on feedback from the previous interviewer. This ‘boosts’ the efficiency of the interview process by deploying a more dynamic evaluation process.
5. **Gradient Boosting:** A special case of boosting where errors are minimized by gradient descent algorithm e.g. the strategy consulting firms leverage by using case interviews to weed out less qualified candidates.

6. **XGBoost:** Think of XGBoost as gradient boosting on ‘steroids’ (well it is called ‘Extreme Gradient Boosting’ for a reason!). It is a perfect combination of software and hardware optimization techniques to yield superior results using less computing resources in the shortest amount of time.

7.3 ALGORITHMIC ENHANCEMENTS:

Regularization: It penalizes more complex models through both LASSO (L1) and Ridge (L2) regularization to prevent overfitting.

1. **Sparsity Awareness:** XGBoost naturally admits sparse features for inputs by automatically ‘learning’ best missing value depending on training loss and handles different types of sparsity patterns in the data more efficiently.
2. **Weighted Quantile Sketch:** XGBoost employs the distributed weighted Quantile Sketch algorithm to effectively find the optimal split points among weighted datasets.
3. **Cross-validation:** The algorithm comes with built-in cross-validation method at each iteration, taking away the need to explicitly program this search and to specify the exact number of boosting iterations required in a single run.

7.4 ADVANTAGES OF XGBOOST ALGORITHM

XGBoost is an efficient and easy to use algorithm which delivers high performance and accuracy as compared to other algorithms. XGBoost is also known as regularized version of GBM. Let see some of the advantages of XGBoost algorithm:

1. **Regularization:** XGBoost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. That is why, XGBoost is also called regularized form of GBM (Gradient Boosting Machine). While using Scikit Learn library, we pass two hyper-parameters (alpha and lambda) to XGBoost related to regularization. alpha is used for L1 regularization and lambda is used for L2 regularization.
2. **Parallel Processing:** XGBoost utilizes the power of parallel processing and that is why it is much faster than GBM. It uses multiple CPU cores to execute the model. While using Scikit Learn library, nthread hyper-parameter is used for parallel processing. nthread represents number of CPU cores to be used. If you want to use all the available cores, don't mention any value for nthread and the algorithm will detect automatically.

3. Handling Missing Values:XGBoost has an in-built capability to handle missing values. When XGBoost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. It then does the same when working on the testing data.

4. Cross Validation:XGBoost allows user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run. This is unlike GBM where we have to run a grid-search and only a limited values can be tested.

5. Effective Tree Pruning:A GBM would stop splitting a node when it encounters a negative loss in the split. Thus it is more of a greedy algorithm.XGBoost on the other hand make splits upto the `max_depth` specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.

7.5 WORKING OF XGBOOST

XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models. When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions). The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. Below is a brief illustration on how gradient tree boosting works.

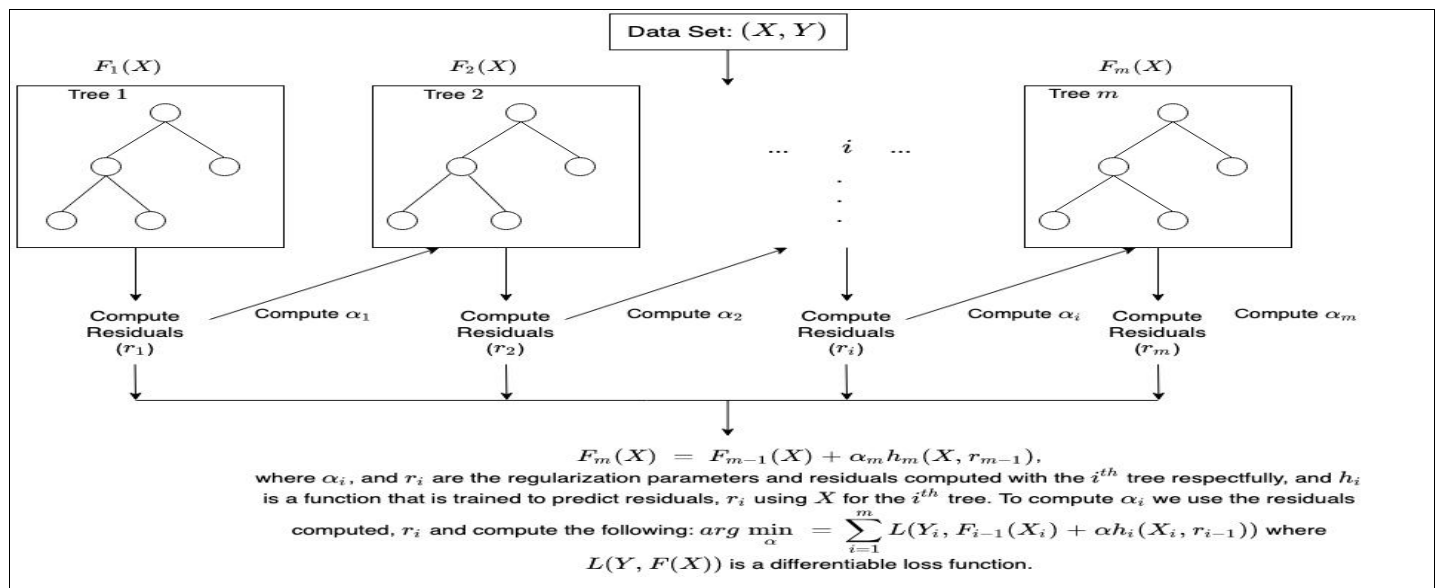


Figure 7.2: Working of gradient tree booting

Decision trees create a model that predicts the label by evaluating a tree of if-then-else true/false feature questions, and estimating the minimum number of questions needed to assess the probability of making a correct decision. Decision trees can be used for classification to predict a category, or regression to predict a continuous numeric value. In the simple example below, a decision tree is used to estimate a house price (the label) based on the size and number of bedrooms (the features).

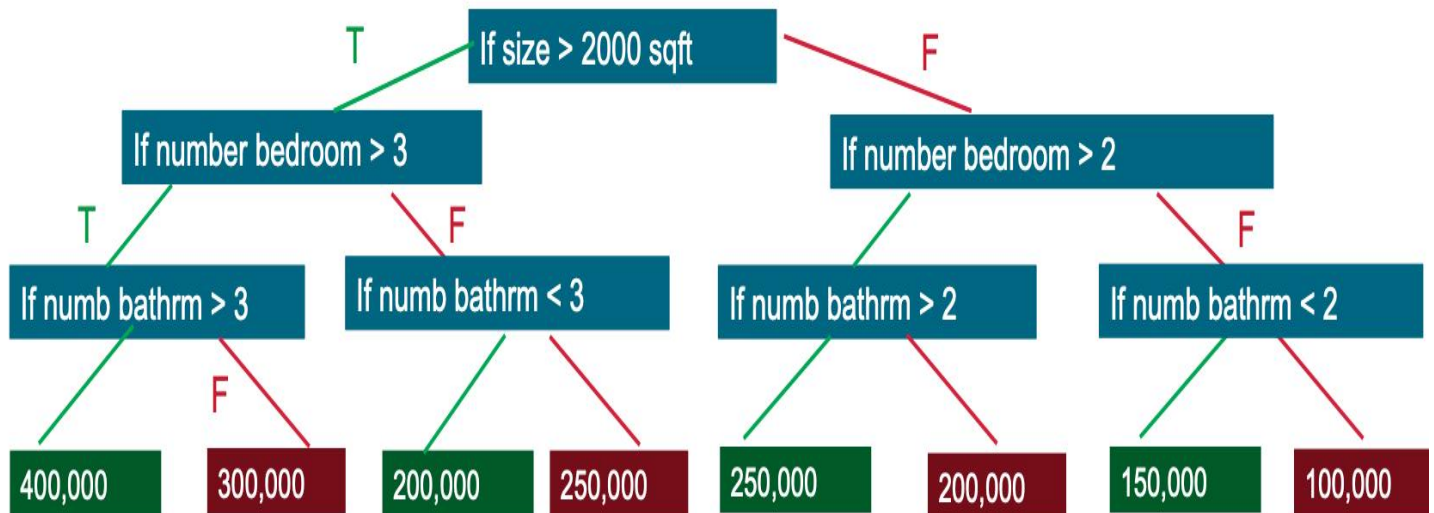


Figure 7.3: Example of working of gradient tree

A Gradient Boosting Decision Trees (GBDT) is a decision tree **ensemble learning algorithm** similar to random forest, for classification and regression. Ensemble learning algorithms combine multiple machine learning algorithms to obtain a better model. Both random forest and GBDT build a model consisting of multiple decision trees. The difference is in how the trees are built and combined.

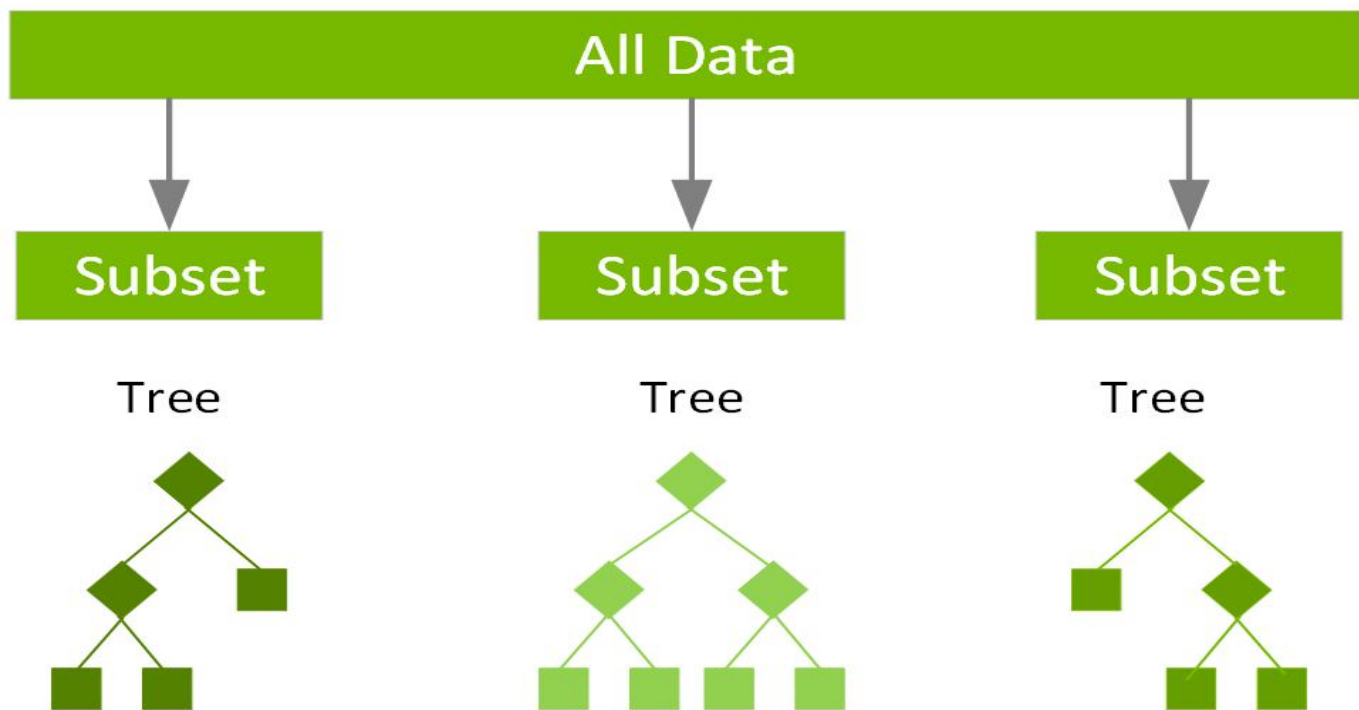


Figure 7.4 : Decision making trees

Random forest uses a technique called bagging to build full decision trees in parallel from random bootstrap samples of the data set. The final prediction is an average of all of the decision tree predictions.

The term “gradient boosting” comes from the idea of “boosting” or improving a single weak model by combining it with a number of other weak models in order to generate a collectively strong model. **Gradient boosting** is an extension of boosting where the process of additively generating weak models is formalized as a gradient descent algorithm over an objective function. Gradient boosting sets targeted outcomes for the next model in an effort to minimize errors. Targeted outcomes for each case are based on the gradient of the error (hence the name gradient boosting) with respect to the prediction.

GBDTs iteratively train an ensemble of shallow decision trees, with each iteration using the error residuals of the previous model to fit the next model. The final prediction is a weighted sum of all of the tree predictions. Random forest “bagging” minimizes the variance and overfitting, while GBDT “boosting” minimizes the bias and underfitting.

XGBoost is a scalable and highly accurate implementation of gradient boosting that pushes the limits of computing power for boosted tree algorithms, being built largely for energizing machine learning model performance and computational speed. With XGBoost, trees are built in parallel, instead of sequentially like GBDT. It follows a level-wise strategy, scanning across gradient values and using these partial sums to evaluate the quality of splits at every possible split in the training set.

7.6 MATHEMATICS BEHIND XGBOOST

Before beginning with mathematics about Gradient Boosting, Here's a simple example of a CART that classifies whether someone will like a hypothetical computer game X. The example of tree is below:

The prediction scores of each individual decision tree then sum up to get If you look at the example, an important fact is that the two trees try to *complement* each other. Mathematically, we can write our model in the form

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F}$$

where, K is the number of trees, f is the functional space of F, F is the set of possible CARTs. The objective function for the above model is given by:

$$obj(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

where, first term is the loss function and the second is the regularization parameter. Now, Instead of learning the tree all at once which makes the optimization harder, we apply the additive strategy, minimize the loss what we have learned and add a new tree which can be summarised below:

$$\begin{aligned} \hat{y}_i^{(0)} &= 0 \\ \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ &\dots \\ \hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \end{aligned}$$

The objective function of the above model can be defined as:

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \\ &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant \\ obj^{(t)} &= \sum_{i=1}^n (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^t \Omega(f_i) \\ &= \sum_{i=1}^n [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant \end{aligned}$$

Now, let's apply Taylor series expansion upto second order:

$$obj^{(t)} = \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + \text{constant}$$

where g_i and h_i can be defined as:

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$

$$h_i = \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)})$$

Simplifying and removing the constant:

$$\sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

Now, we define the regularization term, but first we need to define the model:

$$f_t(x) = w_{q(x)}, w \in R^T, q : R^d \rightarrow \{1, 2, \dots, T\}$$

Here, w is the vector of scores on leaves of tree, q is the function assigning each data point to the corresponding leaf, and T is the number of leaves. The regularization term is then defined by:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

Now, our objective function becomes:

$$obj^{(t)} \approx \sum_{i=1}^n [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

$$= \sum_{j=1}^T [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T$$

Now, we simplify the the above expression:

$$obj^{(t)} = \sum_{j=1}^T [G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2] + \gamma T$$

where,

$$G_j = \sum_{i \in I_j} g_i$$

$$H_j = \sum_{i \in I_j} h_i$$

In this equation, w_j are independent of each other, the best w_j for a given structure $q(x)$ and the best objective reduction we can get is:

$$w_j^* = -\frac{G_j}{H_j + \lambda}$$

$$\text{obj}^* = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T$$

where, γ is pruning parameter, i.e the least information gain to perform split.

Now, we try to measure how good the tree is, we can't directly optimize the tree, we will try to optimize one level of the tree at a time. Specifically we try to split a leaf into two leaves, and the score it gains is

$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma$$

7.7 FLOWCHART

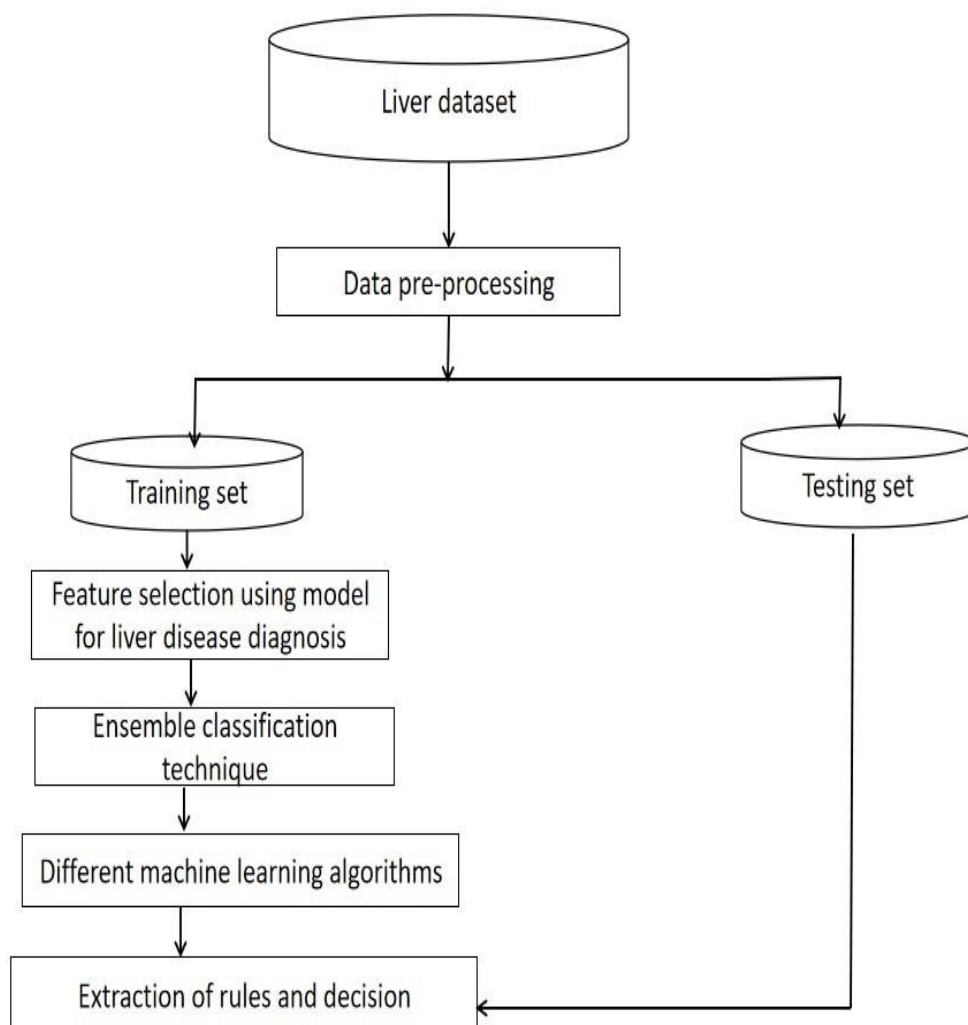


Figure 7.5: Flow chart

CHAPTER 8: GRAPHICAL USER INTERFACE IMPLEMENTATION

8.1 TKINTER PROGRAMMING

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Example

```
#!/usr/bin/python
import Tkinter
top = Tkinter.Tk()
# Code to add widgets will go here...
top.mainloop()
```

This would create a following window –



Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table –

Sr.No.	Operator & Description
1	<p>Button</p> <p>The Button widget is used to display buttons in your application.</p>
2	<p>Canvas</p> <p>The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.</p>
3	<p>Checkbutton</p> <p>The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.</p>
4	<p>Entry</p> <p>The Entry widget is used to display a single-line text field for accepting values from a user.</p>
5	<p>Frame</p> <p>The Frame widget is used as a container widget to organize other widgets.</p>
6	<p>Label</p> <p>The Label widget is used to provide a single-line caption for other widgets. It can also contain images.</p>
7	<p>Listbox</p> <p>The Listbox widget is used to provide a list of options to a user.</p>
8	<p>Menubutton</p> <p>The Menubutton widget is used to display menus in your application.</p>
9	<p>Menu</p>

	<p>The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.</p>
10	<p>Message</p> <p>The Message widget is used to display multiline text fields for accepting values from a user.</p>
11	<p>Radiobutton</p> <p>The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.</p>
12	<p>Scale</p> <p>The Scale widget is used to provide a slider widget.</p>
13	<p>Scrollbar</p> <p>The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.</p>
14	<p>Text</p> <p>The Text widget is used to display text in multiple lines.</p>
15	<p>Toplevel</p> <p>The Toplevel widget is used to provide a separate window container.</p>
16	<p>Spinbox</p> <p>The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.</p>
17	<p>PanedWindow</p> <p>A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.</p>
18	<p>LabelFrame</p>

	<p>A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.</p>
19	<p>tkMessageBox</p> <p>This module is used to display message boxes in your applications.</p>

CHAPTER 9: RESULTS

9.1 EXISTING MODEL ACCURACY

```

memory usage: 7.012 MB
For Fold 1 the accuracy is 0.6904761904761905
For Fold 2 the accuracy is 0.7857142857142857
For Fold 3 the accuracy is 0.6190476190476191
For Fold 4 the accuracy is 0.6666666666666666
For Fold 5 the accuracy is 0.8095238095238095
For Fold 6 the accuracy is 0.6666666666666666
For Fold 7 the accuracy is 0.6904761904761905
For Fold 8 the accuracy is 0.7380952380952381
For Fold 9 the accuracy is 0.7317073170731707
For Fold 10 the accuracy is 0.6341463414634146

Logestic Regression Mean Accuracy = 0.7032520325203252
      precision    recall  f1-score   support

         0         0.70         0.78         0.74          27
         1         0.45         0.36         0.40          14

   accuracy                   0.63          41
  macro avg         0.58         0.57         0.57          41
weighted avg         0.62         0.63         0.62          41

AUC : 0.6507936507936508
    
```

9.2 PROPOSED MODEL ACCURACY

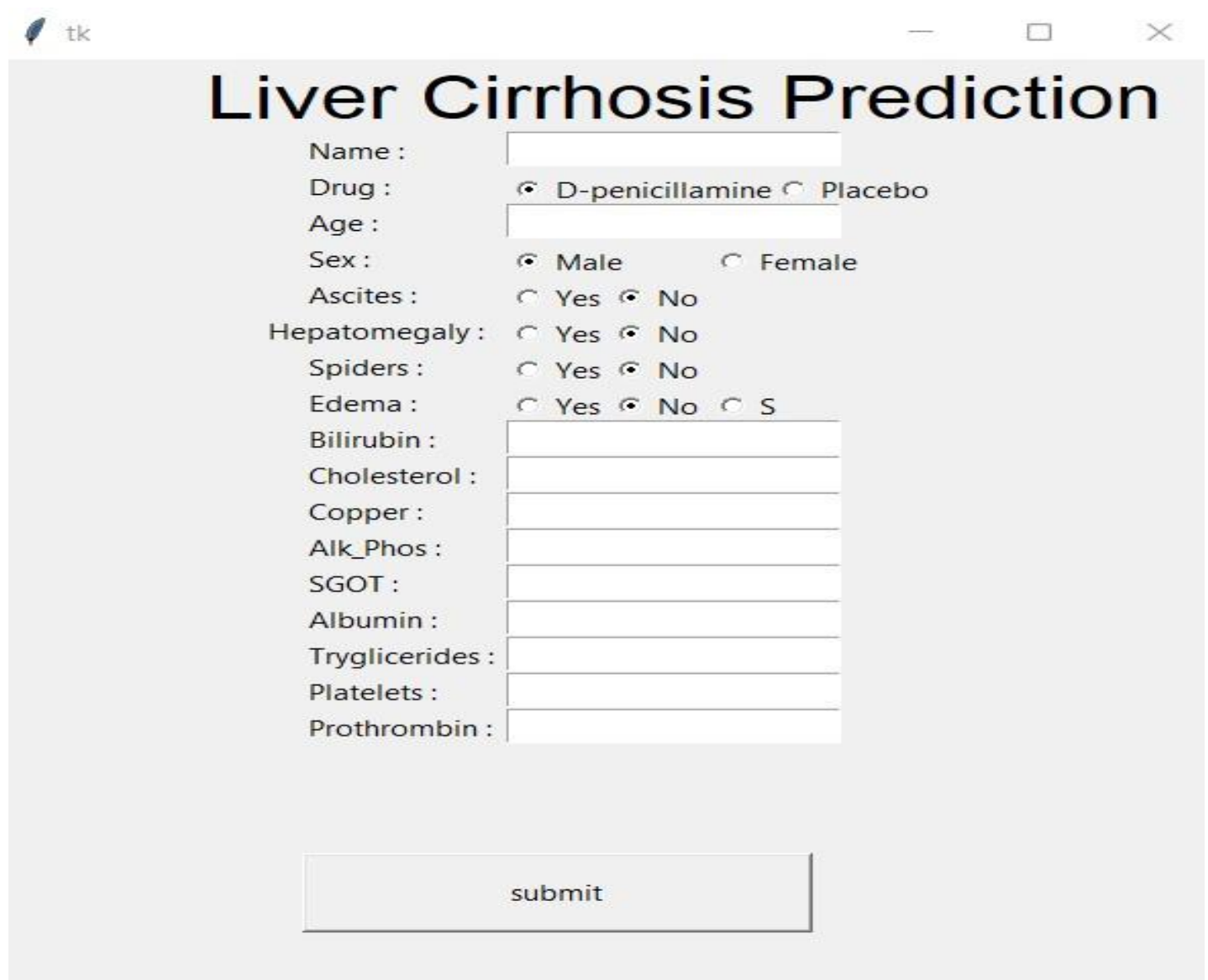
For Fold 10 the accuracy is 0.7804878048780488

XGboost model Mean Accuracy = 0.7344367015098723

	precision	recall	f1-score	support
0	0.82	0.85	0.84	27
1	0.69	0.64	0.67	14
accuracy			0.78	41
macro avg	0.76	0.75	0.75	41
weighted avg	0.78	0.78	0.78	41

AUC : 0.738095238095238

9..3 GRAPHICAL USER INTERFACE



The image shows a graphical user interface (GUI) window titled "Liver Cirrhosis Prediction". The window has a standard Tkinter title bar with a feather icon, the text "tk", and window control buttons (minimize, maximize, close). The main content area is light gray and contains a form with the following fields and controls:

- Name :** A text input field.
- Drug :** A radio button group with two options: ☒ D-penicillamine and ☐ Placebo.
- Age :** A text input field.
- Sex :** A radio button group with two options: ☒ Male and ☐ Female.
- Ascites :** A radio button group with two options: ☐ Yes and ☒ No.
- Hepatomegaly :** A radio button group with two options: ☐ Yes and ☒ No.
- Spiders :** A radio button group with two options: ☐ Yes and ☒ No.
- Edema :** A radio button group with three options: ☐ Yes, ☒ No, and ☐ S.
- Bilirubin :** A text input field.
- Cholesterol :** A text input field.
- Copper :** A text input field.
- Alk_Phos :** A text input field.
- SGOT :** A text input field.
- Albumin :** A text input field.
- Tryglicerides :** A text input field.
- Platelets :** A text input field.
- Prothrombin :** A text input field.

At the bottom of the form is a large rectangular button labeled "submit".

Liver Cirrhosis Prediction

Name :

Drug : ☒ D-penicillamine ☐ Placebo

Age :

Sex : ☒ Male ☐ Female

Ascites : ☐ Yes ☒ No

Hepatomegaly : ☐ Yes ☒ No

Spiders : ☐ Yes ☒ No

Edema : ☐ Yes ☒ No ☐ S

Bilirubin :

Cholesterol :

Copper :

Alk_Phos :

SGOT :

Albumin :

Tryglicerides :

Platelets :

Prothrombin :

submit

warning

!

Enter Valid Bilirubin Value

OK

Liver Cirrhosis Prediction

Name :

Drug : ☒ D-penicillamine ☐ Placebo

Age :

Sex : ☒ Male ☐ Female

Ascites : ☐ Yes ☒ No

Hepatomegaly : ☐ Yes ☒ No

Spiders : ☐ Yes ☒ No

Edema : ☐ Yes ☒ No ☐ S

Bilirubin :

Cholesterol :

Copper :

Alk_Phos :

SGOT :

Albumin :

Tryglicerides :

Platelets :

Prothrombin :

submit

warning

!

All Fields are Required

OK

tk

Liver Cirrhosis Prediction

Name : fsf

Drug : ☒ D-penicillamine ☐ Placebo

Age : 55

Sex : ☒ Male ☐ Female

Ascites : ☐ Yes ☒ No

Hepatomegaly : ☐ Yes ☒ No

Spiders : ☐ Yes ☒ No

Edema : ☐ Yes ☒ No ☐ S

Bilirubin : 0.5

Cholesterol : 1000

Copper : 400

Alk_Phos : 400

SGOT : 400

Albumin : 3.0

Tryglicerides : 525

Platelets : 560

Prothrombin : 15

submit

showinfo

fsf has no liver cirrhosis

OK

tk

Liver Cirrhosis Prediction

Name : fsf

Drug : ☒ D-penicillamine ☐ Placebo

Age : 55

Sex : ☒ Male ☐ Female

Ascites : ☒ Yes ☐ No

Hepatomegaly : ☐ Yes ☒ No

Spiders : ☒ Yes ☐ No

Edema : ☐ Yes ☒ No ☐ S

Bilirubin : 0.5

Cholesterol : 1000

Copper : 400

Alk_Phos : 400

SGOT : 400

Albumin : 3.0

Tryglicerides : 525

Platelets : 560

Prothrombin : 15

submit

showinfo

fsf has liver cirrhosis

OK