**1 INTRODUCTION**

**1.1 INTRODUCTION**

The liver is a large, meaty organ that sits on the right side of the belly. Weighing about 3 pounds, the liver is reddish brown in color and feels rubbery to the feel. The liver has two large sections, called the right and the left lobes. The gallbladder sits below the liver, along with parts of the pancreas and intestines. The liver and these organs behavior together to digest, absorb, and process food. The liver's main job is to strain the blood coming from the digestive tract, before passing it to the rest of the body. The liver also detoxifies chemicals and metabolizes drugs. As it does so, the liver hides bile that ends up back in the intestines. The liver also makes proteins important for blood clotting and other functions.

Liver disease is any trouble of liver function that causes sickness. The liver is responsible for many dangerous functions within the body and should it become diseased or damaged, the loss of those functions can cause significant injury to the body. Liver disease is also referred to as hepatic disease. Liver disease is a large term that covers all the potential problems that cause the liver to fail to perform its designated functions.

**1.2 OBJECTIVE OF THE PROJECT**

This **project Liver Disease Prediction Using Machine Learning** is a machine learning application. In this project, you predict whether the patient contain a liver disease or not using python Jupyter Notebook. To predict presence of liver disease we apply some of the classification techniques.

It gives an idea of how machine learning helps in medical field and how classification techniques going to predict liver disease using liver disease data set.

**1.3 ORGANIZTION OF THE REPORT**

The first chapter deals with introduction of snake game uing, motivation for developing this project, objective of the project. The second chapter deals with the system specifications required for developing the project. It includes hardware & software specifications. The third chapter gives you the Design and implementation which includes introduction, source code, Description of key parameters & functions, methods of implementation, testing and validations. Finally fourth chapter deals with the conclusion and future enhancements.

**2 SYSTEM SPECIFICATION**

**2.1 SOFTWARE SPECIFICATION**

* Operating system : WINDOWS
* Application software : Jupyter Notebook
* Language : python
* Data set : Liver Disease data set downloaded from Kaggle

dataset

**2.2 HARDWARE SPECIFIACTION**

* Hard Disk : 32 GB
* RAM : 4 GB
* Processor : Any Pentium version

**3 DESIGN AND IMPLEMENTATION**

**3.1 INTRODUCTION**

**Machine Learning** is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: **The ability to learn.**

**Machine learning** is the study of computer [algorithms](https://en.wikipedia.org/wiki/Algorithm) that improve automatically through experience. It is seen as a part of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a model based on sample data, known as "[training data](https://en.wikipedia.org/wiki/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 [email filtering](https://en.wikipedia.org/wiki/Email_filtering) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

**3.2 MACHINE LEARNING LIFE CYCLE**

Machine learning life cycle is a cyclic process to build an efficient machine learning project. The main purpose of the life cycle is to find a solution to the problem or project. Machine learning life cycle involves seven major steps, which are given below:

* Data Gathering
* Data preparation
* Data wrangling
* Data Analysis
* Train model
* Test model
* Deployment

**3.3 DATA GATHERING**

Data Gathering is the first step of the machine learning life cycle. The goal of this step is to identify and obtain all data-related problems. In this step, we need to identify the different data sources, as data can be collected from various sources such as **files**, **database**, **internet**, or **mobile devices**. It is one of the most important steps of the life cycle. The quantity and quality of the collected data will determine the efficiency of the output. The more will be the data, the more accurate will be the prediction.

This step includes the below tasks:

* **Identify various data sources**
* **Collect data**
* **Integrate the data obtained from different sources**

By performing the above task, we get a coherent set of data, also called as a **dataset**. It will be used in further steps.

Liver disease data set used in this project is:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| age | gender | Total Bilirubin | Direct Bilirubin | Alkphos Alkaline Phosphotase | Sgpt Alamine Aminotransferase | Sgot Aspartate Aminotransferase | Total Protiens | Albumin | A/G | R |
| 65 | Female | 0.7 | 0.1 | 187 | 16 | 18 | 6.8 | 3.3 | 0.9 | 1 |
| 62 | Male | 10.9 | 5.5 | 699 | 64 | 100 | 7.5 | 3.2 | 0.74 | 1 |
| 62 | Male | 7.3 | 4.1 | 490 | 60 | 68 | 7 | 3.3 | 0.89 | 1 |
| 58 | Male | 1 | 0.4 | 182 | 14 | 20 | 6.8 | 3.4 | 1 | 1 |
| 72 | Male | 3.9 | 2 | 195 | 27 | 59 | 7.3 | 2.4 | 0.4 | 1 |

**3.4 DATA PREPARATION**

After collecting the data, we need to prepare it for further steps. Data preparation is a step where we put our data into a suitable place and prepare it to use in our machine learning training.

In this step, first, we put all data together, and then randomize the ordering of data. This step can be further divided into two processes:

* **Data exploration:**

It is used to understand the nature of data that we have to work with. We need to understand the characteristics, format, and quality of data.A better understanding of data leads to an effective outcome. In this, we find Correlations, general trends, and outliers.

* **Data pre-processing:**  
  Now the next step is pre-processing of data for its analysis.

**3.5 DATA WRANGLING**

Data wrangling is the process of cleaning and converting raw data into a useable format. It is the process of cleaning the data, selecting the variable to use, and transforming the data in a proper format to make it more suitable for analysis in the next step. It is one of the most important steps of the complete process. Cleaning of data is required to address the quality issues.

It is not necessary that data we have collected is always of our use as some of the data may not be useful. In real-world applications, collected data may have various issues, including:

* **Missing Values**
* **Duplicate data**
* **Invalid data**
* **Noise**

So, we use various filtering techniques to clean the data. It is mandatory to detect and remove the above issues because it can negatively affect the quality of the outcome.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| age | gender | Total Bilirubin | Direct Bilirubin | Alkphos Alkaline Phosphotase | Sgpt Alamine Aminotransferase | Sgot Aspartate Aminotransferase | Total Protiens | Albumin | A/G | R |
| 65 | Female | **NAN** | 0.1 | 187 | 16 | 18 | 6.8 | 3.3 | 0.9 | 1 |
| 62 | Male | 10.9 | 5.5 | 699 | 64 | **NAN** | 7.5 | 3.2 | 0.74 | 1 |
| 62 | Male | 7.3 | 4.1 | 490 | 60 | 68 | 7 | 3.3 | 0.89 | 1 |
| 58 | Male | 1 | 0.4 | **NAN** | 14 | 20 | **NAN** | 3.4 | 1 | 1 |
| 72 | Male | 3.9 | 2 | 195 | 27 | 59 | 7.3 | 2.4 | 0.4 | 1 |

**3.6 DATA ANALYSIS**

Now the cleaned and prepared data is passed on to the analysis step. This step involves:

* **Selection of analytical techniques**
* **Building models**
* **Review the result**

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as **Classification** i.e., **logistic regression, decision tree, random forest** etc., then build the model using prepared data, and evaluate the model. Hence, in this step, we take the data and use machine learning algorithms to build the model.

**3.7 TRAIN MODEL**

Now the next step is to train the model, in this step we train our model to improve its performance for better outcome of the problem.

We use datasets to train the model using various machine learning algorithms. Training a model is required so that it can understand the various patterns, rules, and, features.

**3.8 TEST MODEL**

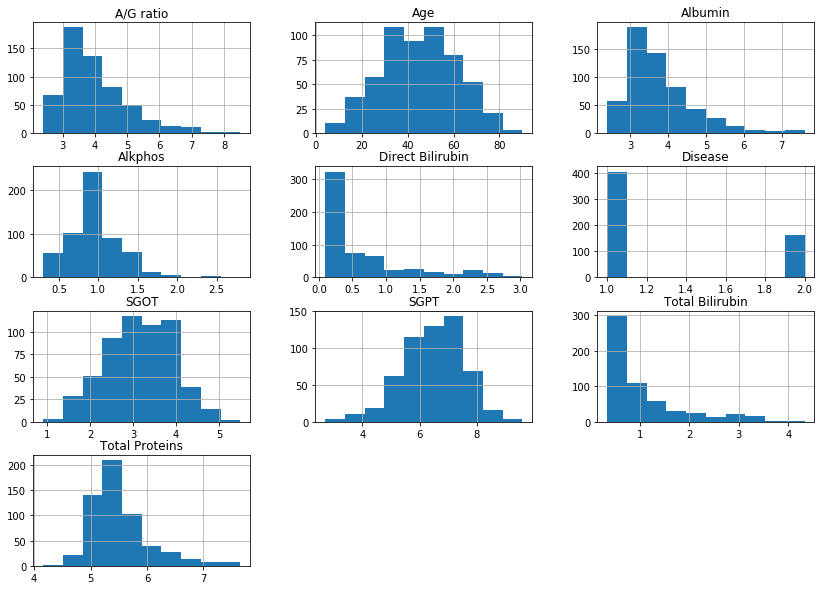
Once our machine learning model has been trained on a given dataset, then we test the model. In this step, we check for the accuracy of our model by providing a test dataset to it.

Testing the model determines the percentage accuracy of the model as per the requirement of project or problem.

**3.9 DEPLOYMENT**

The last step of machine learning life cycle is deployment, where we deploy the model in the real-world system. If the above-prepared model is producing an accurate result as per our requirement with acceptable speed, then we deploy the model in the real system. But before deploying the project, we will check whether it is improving its performance using available data or not. The deployment phase is similar to making the final report for a project.

**3.10 DATA VISUALIZATION**

****

**4 CLASSIFICATION TECHNIQUES**

**4.1 INTRODUCTION**

Classification is defined as the process of recognition, understanding, and grouping of objects and ideas into preset categories “sub-populations.” With the help of these pre-categorized training datasets, classification in machine learning programs leverage a wide range of algorithms to classify future datasets into respective and relevant categories.

Classification algorithms used in machine learning utilize input training data for the purpose of predicting the likelihood or probability that the data that follows will fall into one of the predetermined categories. One of the most common applications of classification is for filtering emails into “spam” or “non-spam”, as used by today’s top email service providers.

In short, classification is a form of “pattern recognition,”. Here, classification algorithms applied to the training data find the same pattern.

Different classification techniques used in this project are:

* Logistic Regression
* Decision Tree
* Random Forest
* Support Vector Machine
* Naïve Bayes

**4.2 LOGISTIC REGRESSION**

* Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
* Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.



* Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic regression is used for solving the classification problems**.
* In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

We know the equation of the straight line can be written as:

Logistic Regression in Machine Learning

In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

Logistic Regression in Machine Learning

But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

Logistic Regression in Machine Learning

The above equation is the final equation for Logistic Regression.

## Type of Logistic Regression:

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".

**4.2.1 Python Implementation Of Logistic Regression**

we will pre-process/prepare the data so that we can use it in our code efficiently. It will be the same as we have done in Data pre-processing topic. The code for this is given below:

1. **#Data Pre-procesing Step**
2. **# importing libraries**
3. **import numpy as nm**
4. **import matplotlib.pyplot as mp**
5. **import pandas as pd**
6. **#importing datasets**
7. **a=pd.read\_excel(r"C:\Users\vennela\Documents\miniproject2/traindata1.xlsx ",header=0)**

By executing the above lines of code, we will get the dataset as the output. Consider the given image:

Age of the patient Gender of the patient Total Bilirubin \

0 65.0 Female 0.7

1 62.0 Male 10.9

2 62.0 Male 7.3

3 58.0 Male 1.0

Direct Bilirubin  Alkphos Alkaline Phosphotase \

0 0.1 187.0

1 5.5 699.0

2 4.1 490.0

3 0.4 182.0

Sgpt Alamine Aminotransferase Sgot Aspartate Aminotransferase \

0 16.0 18.0

1 64.0 100.0

2 60.0 68.0

3 14.0 20.0

Total Protiens  ALB Albumin A/G Ratio Albumin and Globulin Ratio \

0 6.8 3.3 0.90

1 7.5 3.2 0.74

2 7.0 3.3 0.89

3 6.8 3.4 1.00

Result

0 1

1 1

2 1

3 1

The categorical values present in the gender column is replaced with female as 0 and male as 1. Below is the code for it:

**a["Gender of the patient"]=a["Gender of the patient"].replace("Female",0)**

**a["Gender of the patient"]=a["Gender of the patient"].replace("Male",1)**

Checking and removing the null values present in the dataset. Below is the code for it:

**print(a.isnull().sum())**

Age of the patient 2

Gender of the patient 902

Total Bilirubin 648

Direct Bilirubin 561

 Alkphos Alkaline Phosphotase 796

 Sgpt Alamine Aminotransferase 538

Sgot Aspartate Aminotransferase 462

Total Protiens 463

 ALB Albumin 494

A/G Ratio Albumin and Globulin Ratio 559

Result 0

dtype: int64

**a=a.dropna()**

**print(a.isnull().sum())**

Age of the patient 0

Gender of the patient 0

Total Bilirubin 0

Direct Bilirubin 0

 Alkphos Alkaline Phosphotase 0

 Sgpt Alamine Aminotransferase 0

Sgot Aspartate Aminotransferase 0

Total Protiens 0

 ALB Albumin 0

A/G Ratio Albumin and Globulin Ratio 0

Result 0

dtype: int64

Now, we will extract the dependent and independent variables from the given dataset. Below is the code for it:

**#Extracting Independent and dependent Variable**

**x=a.iloc[:,2:-1]**

**y=np.array(a.iloc[:,10:])**

In the above code, we have taken [2, 3, 4, 5, 6, 7, 8, 9] for x because our independent

Variables are **Total Bilirubin, Direct Bilirubin, Alkphos Alkaline Phosphotase , Sgpt Alamine Aminotransferase, Sgot Aspartate Aminotransferase, Total Protiens, ALB Albumin A/G Ratio Albumin and Globulin Ratio** , which are at index 2, 3, 4, 5, 6, 7, 8, 9. And we have taken 10 for y variable because our dependent variable is at index 10.

Now we will split the dataset into a training set and test set. Below is the code for it:

1. **# Splitting the dataset into training and test set.**
2. **from sklearn.model\_selection import train\_test\_split**
3. **X\_train, X\_test, Y\_train, Y\_test= train\_test\_split(x, y, random\_state=0)**

In logistic regression, we will do feature scaling because we want accurate result of predictions. Here we will only scale the independent variable because dependent variable have only 0 and 1 values. Below is the code for it:

1. **#feature Scaling**
2. **from sklearn.preprocessing import StandardScaler**
3. **st\_x= StandardScaler()**
4. **x\_train= st\_x.fit\_transform(x\_train)**
5. **x\_test= st\_x.transform(x\_test)**

**Fitting Logistic Regression to the Training set:**

We have well prepared our dataset, and now we will train the dataset using the training set. For providing training or fitting the model to the training set, we will import the **LogisticRegression** class of the **sklearn** library.

After importing the class, we will create a classifier object and use it to fit the model to the logistic regression. Below is the code for it:

1. **#Fitting Logistic Regression to the training set**
2. **from sklearn.linear\_model import LogisticRegression**
3. **classifier= LogisticRegression(random\_state=0)**
4. **classifier.fit(x\_train, y\_train)**

By executing the above code, we will get the below output:

**LogisticRegression(C=1.0, class\_weight=None, dual=False, fit\_intercept=True                      ,intercept\_scaling=1, l1\_ratio=None, max\_iter=100,**

**multi\_class='warn', n\_jobs=None, penalty='l2',**

**random\_state=0, solver='warn', tol=0.0001, verbose=0,                    warm\_start=False)**

**Predicting the Test Result**

Our model is well trained on the training set, so we will now predict the result by using test set data. Below is the code for it:

1. **#Predicting the test set result**
2. **Predicted\_y= classifier.predict(x\_test)**

In the above code, we have created a predicted\_y vector to predict the test set result.

By executing the above code, a new vector (prediction\_y) will be created under the variable explorer option. It can be seen as:

**[1 1 1 ... 1 1 1]**

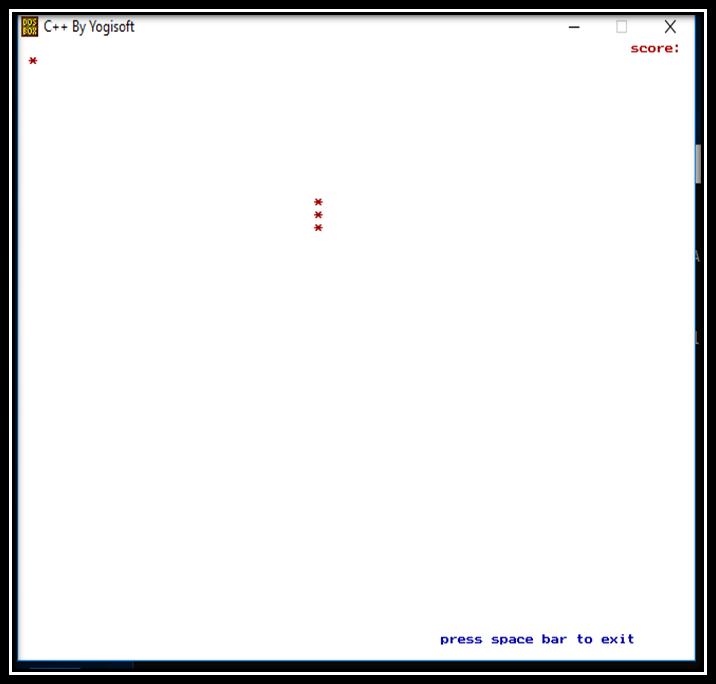
**Test Accuracy of the result**

Now we will create the confusion matrix here to check the accuracy of the classification. To create it, we need to import the **confusion\_matrix** function of the sklearn library. After importing the function, we will call it using a new variable **cm**. The function takes two parameters, mainly **y\_true**( the actual values) and **y\_pred** (the targeted value return by the classifier). Below is the code for it:

**3.8 TESTING AND VALIDATION**

**TESTING**

Testing is the major control measure used during software development. Its basic function is to detect errors in the software. During requirement analysis and design, the output is a document that is usually textual and no executable. After the coding phase, computer programs are available that can be executed for testing purpose. This implies that testing not only, has to uncover errors introduced during coding, but also errors introduced during previous phase. Thus the goal of testing is to uncover the requirements, design and coding errors in the programs. So after testing the outputs of my project are as follows:

****

**Fig 3.6.1** The outlook of the screen

**FUTURE SCOPE OF THE PROJECT**

Our project will be able to implement in future after making some changes and modifications as we make our project at a very low lwvwl. So the modifications that can be done in our project are:

1. The algorithms used in this project work bit a slow. We need to improve the algorithm.
2. More applications can be added to the current applications.
3. The program can be made user friendly by adding more help options.
4. The display of the current can be improved.
5. Images or pictures drawn can be solved.
6. The algorithm if written in languages other than C can make us add more applications easily.
7. We may extend it to draw 3D images.
8. We can add more levels.

**4.CONCLUSION**

**4.1 CONCLUSION**

This project has been successfully implemented using c graphics. Finally we can able to create a game by using c graphics which is more fun to play a game in the same aspect to create it too. By making this project one can able to better understand the logic and more about the functions of c graphics. With this project one can learn the gaming objectives. The player can get away from tension and one can get more fun by playing this snake game. By doing this project one can learn about new functions and logics in c .

**REFERENCES**

**References:**

* **https://www.geeksforgeeks.org**
* [**https://www.youtube.com**](https://www.youtube.com)
* [**https://www.cs.bham.ac.uk**](https://www.cs.bham.ac.uk)
* [**https://www.programmingsimplified.com/c/graphics.h**](https://www.programmingsimplified.com/c/graphics.h)
* **BOOKS**

1. **Let us C by Yashwanth Kanetkar**
2. **Programming with C by E Balaguruswamy**
3. **Graphics under C by Yashwanth Kanetkar**