A

Major Project

On

MACHINE LEARNING MODEL TO DETECT DISEASES IN LIVER

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

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

BY

M.Venkat(187R1A05F6)

K.Thanuja(187R1A05G0)

M.Shivani(187R1A05G3)

Under the Guidance of

Dr. Punyaban Patel

(Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**MACHINE LEARNING MODEL TO DETECT DISEASES IN LIVER**" being submitted by **M.VENKAT(187R1A05F6), KOMMISETTY THANUJA(187R1A05G0) and MADIPEDDI SHIVANI(187R1A05G3)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. Punyaban Patel
Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ____________________

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide **Dr. Punyaban Patel,** Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Mr. A. Uday Kiran, Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Mrs. G. Latha, Mr. A. Kiran Kumar** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

M.VENKAT(187R1A05F6)
K.THANUJA(187R1A05G0)
M.SHIVANI(187R1A05G3)

ABSTRACT

During the recent decades, the risk of Liver disease in people is increasing at a rapid rate and is sought to be one of the fatal diseases in the world. It’s quite a difficult task for researchers to predict the disease from humongous medical databases. To combat this issue, they have come up with machine learning techniques like classification and clustering. The main aim of this project is to predict the chances of a patient having a liver disease using the classification algorithms. These algorithms are compared based on their classification accuracy and execution time. With these performance factors taken into consideration, the algorithm which serves as a better classifier is chosen. Finally, a GUI (Graphical User Interface), used as a medical tool by hospitals is implemented in accordance with the best classifier.

LIST OF FIGURES / TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as “Liver Disease Prediction”. This software provides facility to upload the details and get to know the prediction for Liver disease. This project uses Machine Learning algorithms to classify whether the liver condition is normal. We use KNN,ANN,SVM and Logistic Regression models for the prediction.

1.2 PROJECT PURPOSE

This model will be useful for health industries who need to predict the diseases. The model will be helpful to know whether the liver condition is normal or abnormal using the blood reports of the patient. This information regarding the patients will be helpful for the medical companies in the process.

1.3 PROJECT FEATURES

Plotting of various graphs which help analyse the data completely and have good understanding regarding the data. The accuracy of the model can be predicted using the confusion matrix . Feature engineering is done to understand the rate of importance of the words in the posts.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

2.1 PROBLEM DEFINITION

Liver can be infected by parasites, viruses which cause inflammation and diminish its function. It has the potential to maintain the customary function, even when a part of it is damaged. However, it is important to diagnose liver disease early which can increase the patient’s survival rate. Expert physicians are required for various examination tests to diagnose the liver disease, but it cannot assure the correct diagnosis.

2.2 EXISTING SYSTEM

The existing models of Liver Disease Prediction involves various machine learning classifiers which yield outputs of less accuracy and can’t handle large bundles of data. The poor performance in the training and testing of the liver disorder dataset as resulted from an insufficiency in the dataset. These previously designed systems have been adequate but more works has to be done on their recognition rate for better accuracy in the diagnosis of the liver disease.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- The accuracy of the model is very low .
- The attributes of the dataset are inadequate .
- No proper GUI used as a screening tool for diagnosis.

To avoid all these limitations and make the working more accurately the system needs to be implemented efficiently.

2.3 PROPOSED SYSTEM

The system being proposed here uses concept of machine learning, and the models are first trained, then tested. Finally the most accurate model will predict the final result. At first, the system asks you to enter your details including age, gender, total Bilirubin, direct Bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT and Alkphos. Values of last eight parameters mentioned here, can be known by blood test report of the user. After taking these inputs from the user, the system compares the data input with the training dataset of most accurate model and then predicts the result accordingly as risk or no risk. The algorithms used are Logistic Regression, K-Nearest Neighbor(KNN), Artificial Neural network(ANN) and Support Vector Machine(SVM). The dataset used is The Indian Liver Patient Dataset (ILPD) which was selected from UCI Machine learning repository for this study. It is a sample of the entire Indian population collected from Andhra Pradesh region and comprises of 585 patients data .

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- Ensure data accuracy.
- Minimum training time needed.
- Greater efficiency.
- No medical expertise required.
- Immediate results.

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis.

- Economic Feasibility
- Technical Feasibility
- Behavioral Feasibility

2.4.1 ECONOMIC FEASIBILITY

Development of this application is highly economically feasible. The organization needed not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding resources. Even after the development, the organization will not be in condition to invest more in the organization. Therefore, the system is economically feasible.

2.4.2 TECHNICAL FEASIBILITY

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

2.4.3 BEHAVIOURAL FEASIBILITY

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor: Intel i3
- Hard disk: 16GB and above
- Input devices: Keyboard, mouse
- Ram: 4GB

2.5.2 SOFTWARE REQUIREMENTS

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system: Windows 7,8,10,11
- Coding language: Python
- Tool: Google colab/ jupyter notebook/pycharm

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

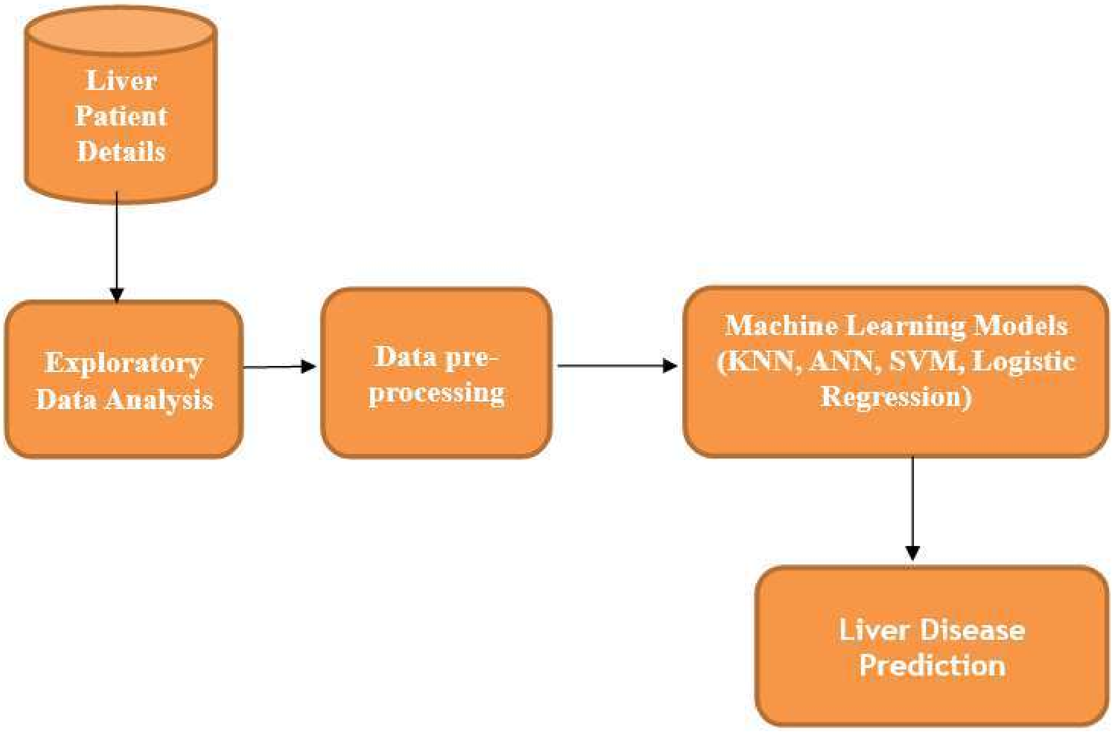

Liver Patient Details
Exploratory Data Analysis
Data pre-processing
Machine Learning Models (KNN, ANN, SVM, Logistic Regression)
Liver Disease Prediction

Figure3.1: Project Architecture for Liver Disease Prediction Model

3.2 MODULE DESCRIPTION

The project is divided into 5 modules. The modular approach of the project is shown below in sequential manner

Module 1: Exploratory Data Analysis

- **Data visualization**: With the help of data visualization, we can see how the data looks like and what kind of correlation is held by the attributes of data. It is the fastest way to see if the features correspond to the output.
- **Features Correlation Analysis**: Correlations have three important characteristics. They can tell us about the direction of the relationship, the form (shape) of the relationship, and the degree (strength) of the relationship between two variables.

Module 2: Preprocessing Stage

- **Eliminating the null and most common words from the text**: The words in the dataset consists of links, multiple full stops, very long and short words. These all need to be eliminated before providing it to the algorithm.

Module 3: Feature Engineering

Splitting into X and Y feature

Hence we split the features as:

X: Bilirubin, phosphate ratio, potassium ratio Y: Liver Condition.

- **Label Encoder** : Provided by Sklearn library that converts the levels of categorical features (labels) into numeric form so as to convert it into the machine-readable form. It encodes labels with a value between 0 and n_classes-1 where n is the number of distinct labels. If a label repeats it assigns the same value to as assigned earlier. We choose label encoding over one-hot encoding to reduce the pre-processing time, and majority due to the fact that there are predefined values under gender and smoker. It seems like a better option to the curse of dimensionality in the feature space.
- **Count Vectorizer** is used to convert a collection of text documents to a vector of term/token counts and build a vocabulary of known words, but also to encode new documents using that vocabulary. It also enables the pre-processing of text data prior to generating the vector representation.

Here, we use stop_words='english' with CountVectorizer since this just counts the occurrences of each word in its vocabulary, extremely common words like ‘the’, ‘and’, etc. will become very important features while they add little meaning to the text. This is an important step in pre-processing

Module 4: Training Classification Model

We split the dataset into testing and training in multiple ratios to give the best results. Now we train the model using the Machine Learning algorithms namely: Logistic Regression, KNN, ANN and SVM to predict the exact result.

LOGISTIC REGRESSION:

Logistic regression is one of the simpler classification models. Because of its parametric nature it can to some extent be interpreted by looking at the parameters making it useful when experimenters want to look at relationships between variables. A parametric model can be described entirely by a vector of parameters = (0, 1... p). An example of a parametric model would be a straight-line y = kx + m where the parameters are k and m. With known parameters the entire model can be recreated. Logistic regression is a parametric model where the parameters are coefficients to the predictor variables written as 0 +1 +X1 + ...PXp Where 0 is called the intercept. For convenience we instead write the above sum of the parameterized predictor variables in vector form as X.

The name logistic regression is a bit unfortunate since a regression model is usually used to find a continuous response variable, whereas in classification the response variable is discrete. The term can be motivated by the fact that we in logistic regression found the probability of the response variable belonging to a certain class. The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE). Once the optimal coefficient (or coefficients if there is more than one independent variable) is found, the conditional probabilities for each observation can be calculated, logged, and summed together to yield a predicted probability. For binary classification, a probability less than .5 will predict 0 while a probability greater than 0 will predict 1. After the model has been computed, it’s best practice to evaluate the how well the model predicts the dependent variable, which is called goodness of fit.

K-NEAREST NEIGHBOR(KNN):

KNN This section describes the implementation details of KNN algorithm. The model for K-Nearest Neighbor is the entire training dataset. When a prediction is required for a unseen data instance, the KNN algorithm will search through the training dataset for the k-most similar instances. The prediction attribute of the most similar instances is summarized and returned as the prediction for the unseen instance. The KNN algorithm is an extreme form of instance-based methods because all training observations are retained as part of the model. It is a competitive learning algorithm, because it internally uses competition between model elements (data instances) in order to make a predictive decision. The objective similarity measure between data instances causes each data instance to compete to win or be most similar to a given unseen data instance and contribute to a prediction.

For classification problems, a class label is assigned on the basis of a majority vote-i.e. the label that is most frequently represented around a given data point is used. While this is technically considered “plurality voting”, the term, “majority vote” is more commonly used in literature. The distinction between these terminologies is that “majority voting” technically requires a majority of greater than 50%, which primarily works when there are only two categories. When you have multiple classes-e.g. four categories, you don’t necessarily need 50% of the vote to make a conclusion about a class; you could assign a class label with a vote of greater than 25%.

ARTIFICIAL NEURAL NETWORK(ANN):

A back propagation neural network was designed. In this network, 10 input neurons were present at the input layer. The number of inputs represents the total number of attributes in the dataset. The input layer uses Rectified Linear Unit activation function. The output layer contains a single layer which uses the sigmoid activation function. In order to obtain a required recognition rate that is capable enough to diagnose the liver disorder in a patient. There is a need for varying certain parameters in the neural network models to produce the required optimum result. These parameters are the learning rate, momentum rate and the hidden neurons. All these parameters present in the backpropagation neural networks. The learning rate is the learning power of the system, the momentum rate determines the learning speed of the system. The number of hidden neurons in the network has to be varied to produce the optimal result. The numbers of neurons needed at the hidden layer are experimenting in order to deter-mine the best neurons that can represent the features present in the input dataset accurately to produce the required optimum result. The numbers of neurons required in the hidden layer were experimenting by varying the neurons. The sigmoid function was used in the output layer because of its soft switching ability and simplicity in derivatives.

At First, information is feed into the input layer which then transfers it to the hidden layers, and interconnection between these two layers assign weights to each input randomly at the initial point. and then bias is added to each input neuron and after this, the weighted sum which is a combination of weights and bias is passed through the activation function. Activation Function has the responsibility of which node to fire for feature extraction and finally output is calculated. This whole process is known as Foreward Propagation. After getting the output model to compare it with the original output and the error is known and finally, weights are updated in backward propagation to reduce the error and this process continues for a certain number of epochs (iteration). Finally, model weights get updated and prediction is done.

SUPPORT VECTOR MACHINE(SVM):

SVM aims to find an optimal hyperplane that separates the data into different classes. The scikit-learn package in python is used for implementing SVM. The pre-processed data is split into test data and training set which is of 25% and 75% of the total dataset respectively. A support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space. A good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3**.**

Module 5: Result Evaluation

Our main goal going into this project was to predict liver disease using various machine learning techniques. We predicted using Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbor (K-NN) and Neural Network. All of them predicted with better results. With Each algorithm, we have observed Accuracy, Precision, Sensitivity and Specificity which can be defined as follows:

Accuracy: The accuracy of a classifier is the percentage of the test set tuples that are correctly classified by the classifier.

$$Accuracy = \frac{no.of\ TP + no.of\ TN}{no.of\ TP + FP + FN + TN}$$

Sensitivity: Sensitivity is also referred as True positive rate i.e. the proportion of positive tuples that are correctly identified.

$$Sensitivity = \frac{no.of\ TP}{no.of\ TP + no.of\ FN}$$

Precision: precision is defined as the proportion of the true positives against all the positive results (both true positives and false positives).

$$Precision = \frac{no.of\ TP}{no.of\ TP + FP}$$

Specificity: Specificity is the True negative rate that is the proportion of negative tuples that are correctly identified.

$$Specificity = \frac{no.of\ TN}{no.of\ TN + FP}$$

The results of each of the classification algorithm is summarized in the table shown below.

Classification Algorithm	**Accuracy**	**Precision**	**Sensitivity**	**Specificity**
Logistic Regression	73.23	78.57	88.00	30.62
K-NN	72.05	80.98	83.78	44.04
SVM	75.04	77.09	79.00	71.11
ANN	92.8	93.78	97.23	83.00

As clearly summarized in the table, Artificial Neural Networks gave the best results.

3.3 USE CASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

An actor is represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

Contents:

- Use cases
- Actors
- Dependency, Generalization, and association relationships
- System boundary

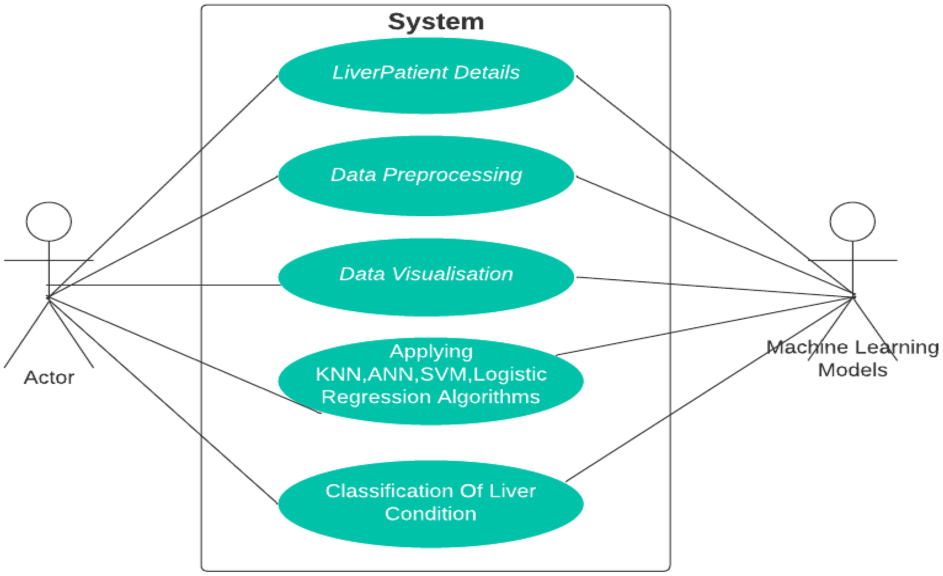


Figure3.2 Usecase diagram for Liver Disease Prediction Model

3.4 CLASS DIAGRAM

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. Class diagrams are arguably the most used UML diagram type. It is the main building block of any object oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class.

In most modeling tools a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many classes related classes are grouped together to create class diagrams. Different relationships between diagrams are show by different types of Arrows. Below is an image of a class diagram

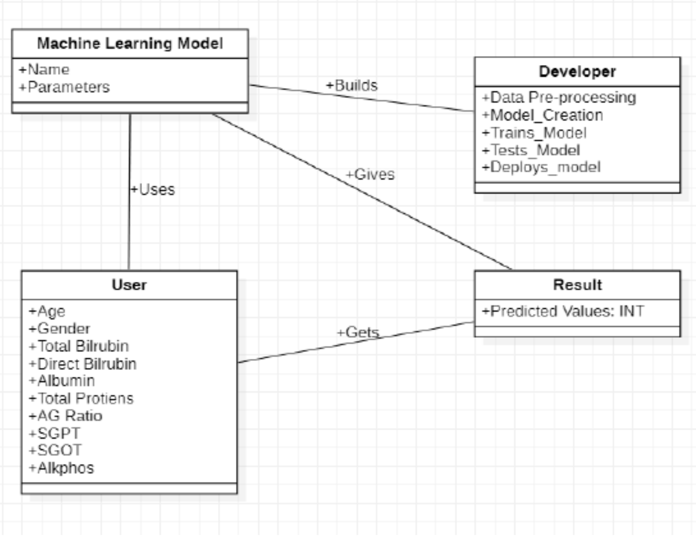


Figure3.3 Class diagram for Liver Disease Prediction Model

3.5 SEQUENCE DIAGRAM

Sequence diagrams in UML shows how object interact with each other and the order those interactions occur. It’s important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are shown as arrows.

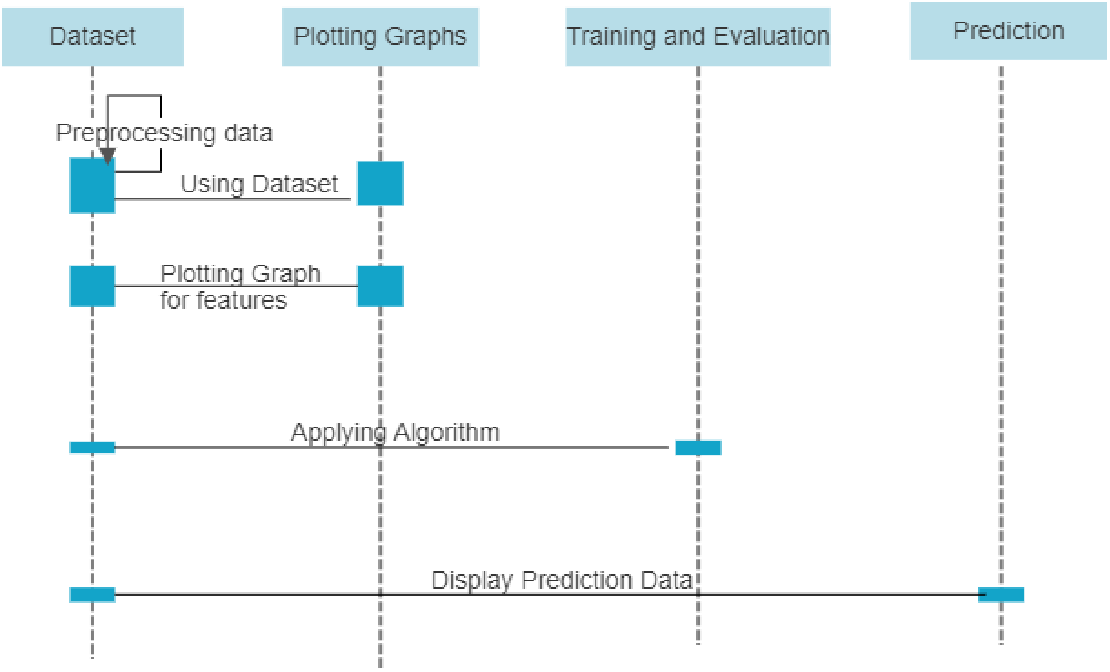


Figure3.4 Sequence diagram for Liver Disease Prediction Model

3.6 ACTIVITY DIAGRAM

It describes about flow of activity states.

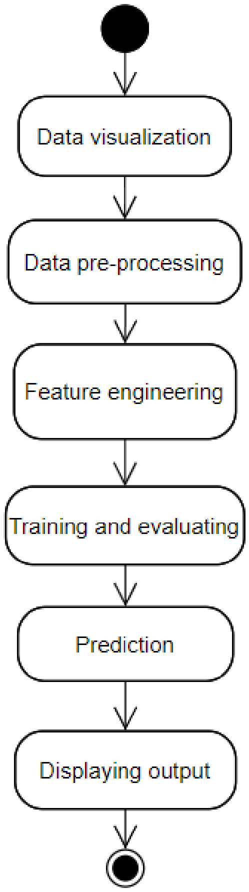
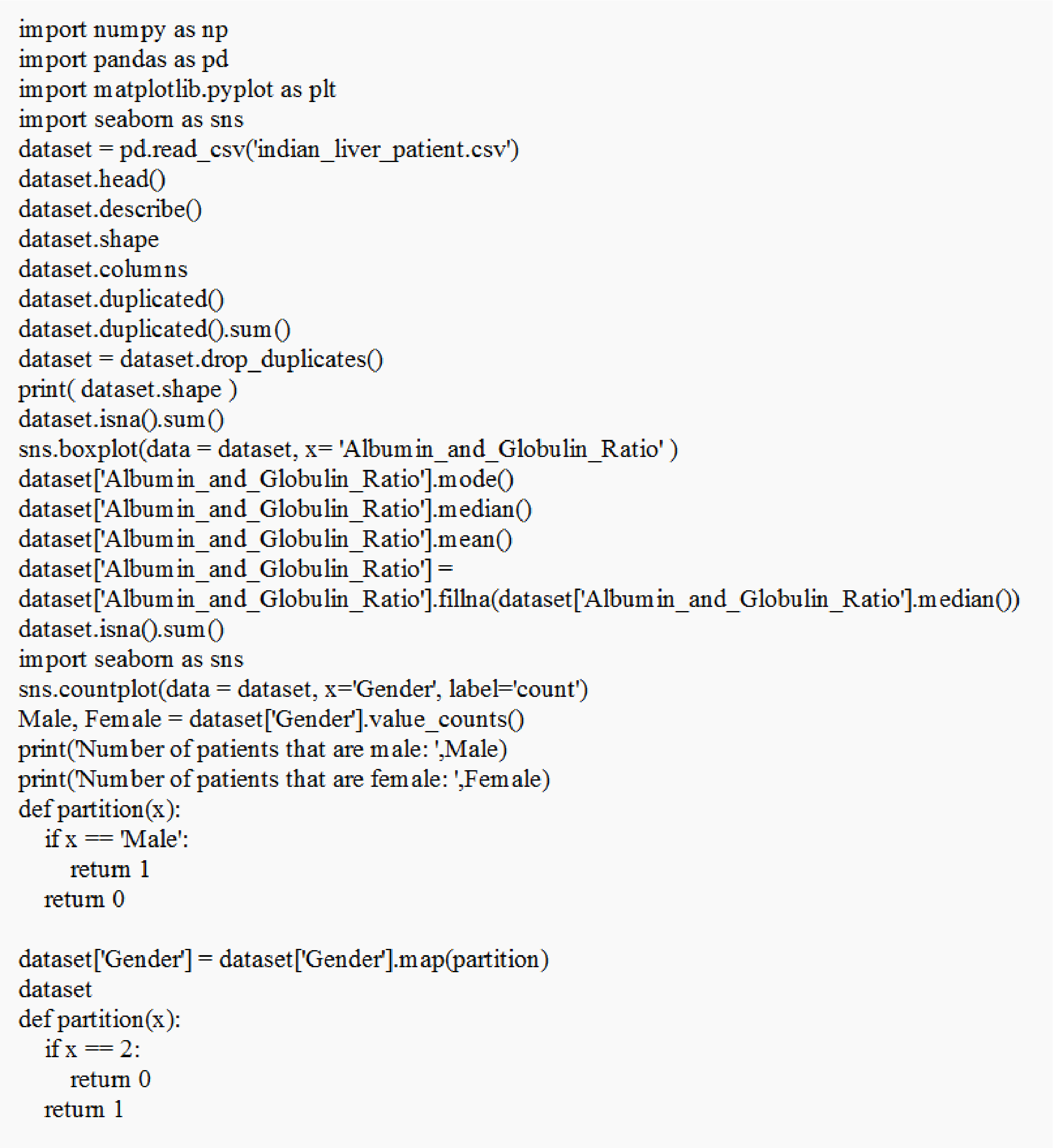


Figure3.5: Activity diagram for Liver Disease Prediction Model

4. IMPLEMENTATION

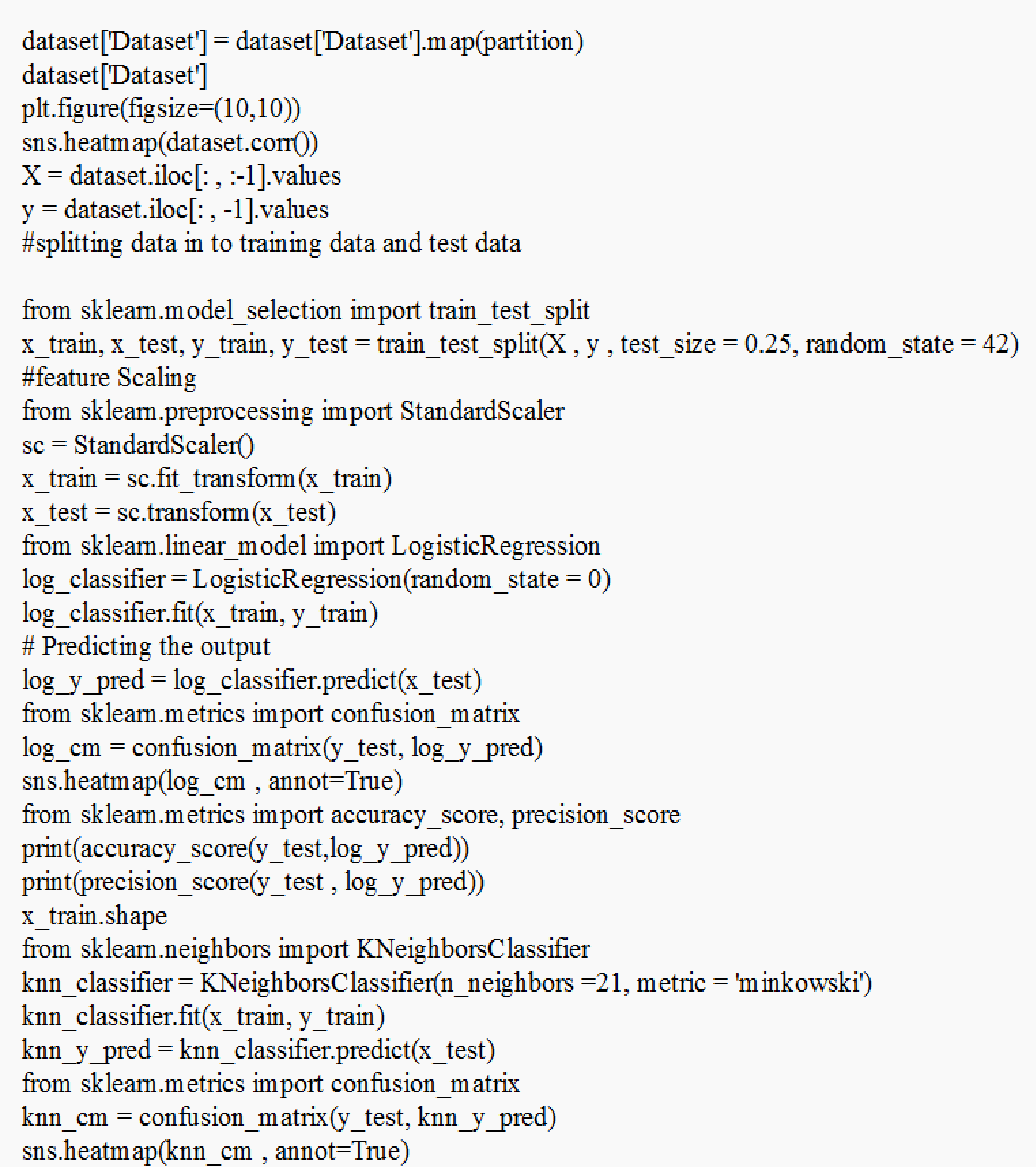
4. IMPLEMENTATION

4.1 SAMPLE CODE



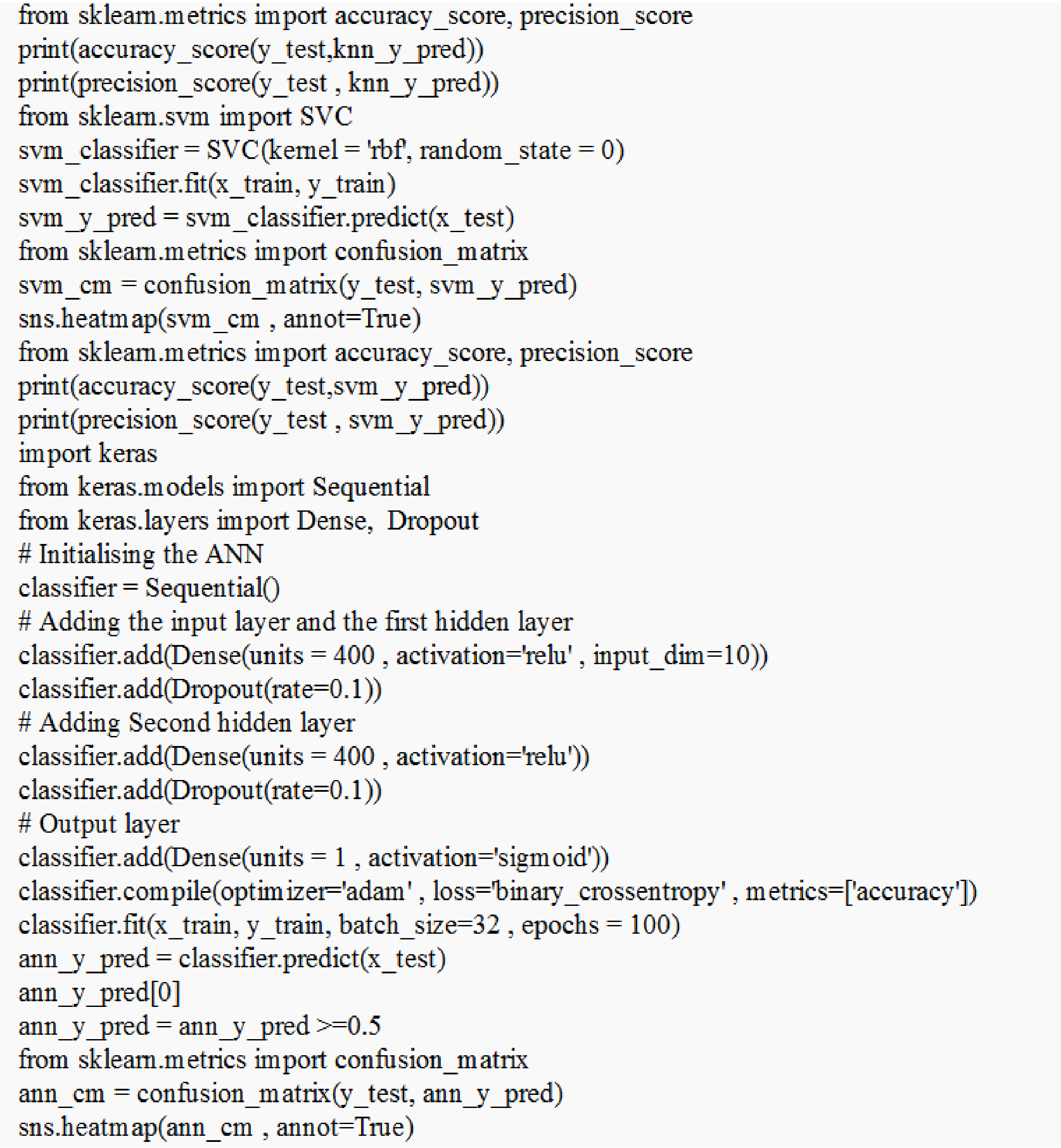
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
dataset = pd.read_csv('indian_liver_patient.csv')
dataset.head()
dataset.describe()
dataset.shape
dataset.columns
dataset.duplicated()
dataset.duplicated().sum()
dataset = dataset.drop_duplicates()
print( dataset.shape )
dataset.isna().sum()
sns.boxplot(data = dataset, x= 'Albumin_and_Globulin_Ratio' )
dataset['Albumin_and_Globulin_Ratio'].mode()
dataset['Albumin_and_Globulin_Ratio'].median()
dataset['Albumin_and_Globulin_Ratio'].mean()
dataset['Albumin_and_Globulin_Ratio'] =
dataset['Albumin_and_Globulin_Ratio'].fillna(dataset['Albumin_and_Globulin_Ratio'].median())
dataset.isna().sum()
import seaborn as sns
sns.countplot(data = dataset, x='Gender', label='count')
Male, Female = dataset['Gender'].value_counts()
print('Number of patients that are male: ',Male)
print('Number of patients that are female: ',Female)
def partition(x):
    if x == 'Male':
        return 1
    return 0

dataset['Gender'] = dataset['Gender'].map(partition)
dataset
def partition(x):
    if x == 2:
        return 0
    return 1
```

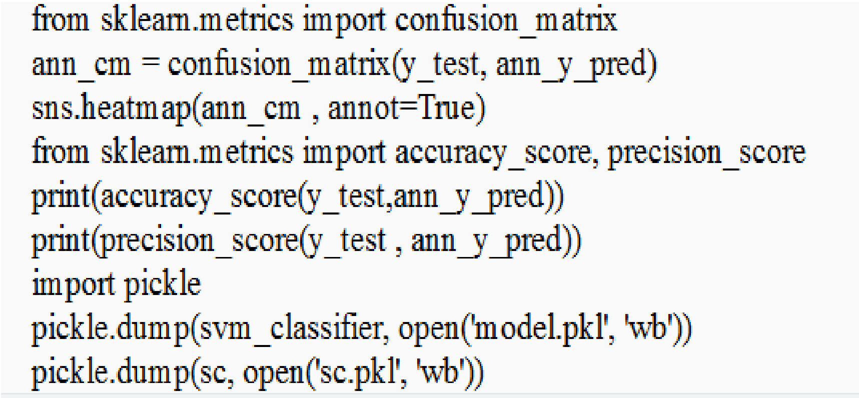


```
dataset['Dataset'] = dataset['Dataset'].map(partition)
dataset['Dataset']
plt.figure(figsize=(10,10))
sns.heatmap(dataset.corr())
X = dataset.iloc[: , :-1].values
y = dataset.iloc[: , -1].values
#splitting data in to training data and test data

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X , y , test_size = 0.25, random_state = 42)
#feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
from sklearn.linear_model import LogisticRegression
log_classifier = LogisticRegression(random_state = 0)
log_classifier.fit(x_train, y_train)
# Predicting the output
log_y_pred = log_classifier.predict(x_test)
from sklearn.metrics import confusion_matrix
log_cm = confusion_matrix(y_test, log_y_pred)
sns.heatmap(log_cm , annot=True)
from sklearn.metrics import accuracy_score, precision_score
print(accuracy_score(y_test,log_y_pred))
print(precision_score(y_test , log_y_pred))
x_train.shape
from sklearn.neighbors import KNeighborsClassifier
knn_classifier = KNeighborsClassifier(n_neighbors =21, metric = 'minkowski')
knn_classifier.fit(x_train, y_train)
knn_y_pred = knn_classifier.predict(x_test)
from sklearn.metrics import confusion_matrix
knn_cm = confusion_matrix(y_test, knn_y_pred)
sns.heatmap(knn_cm , annot=True)
```

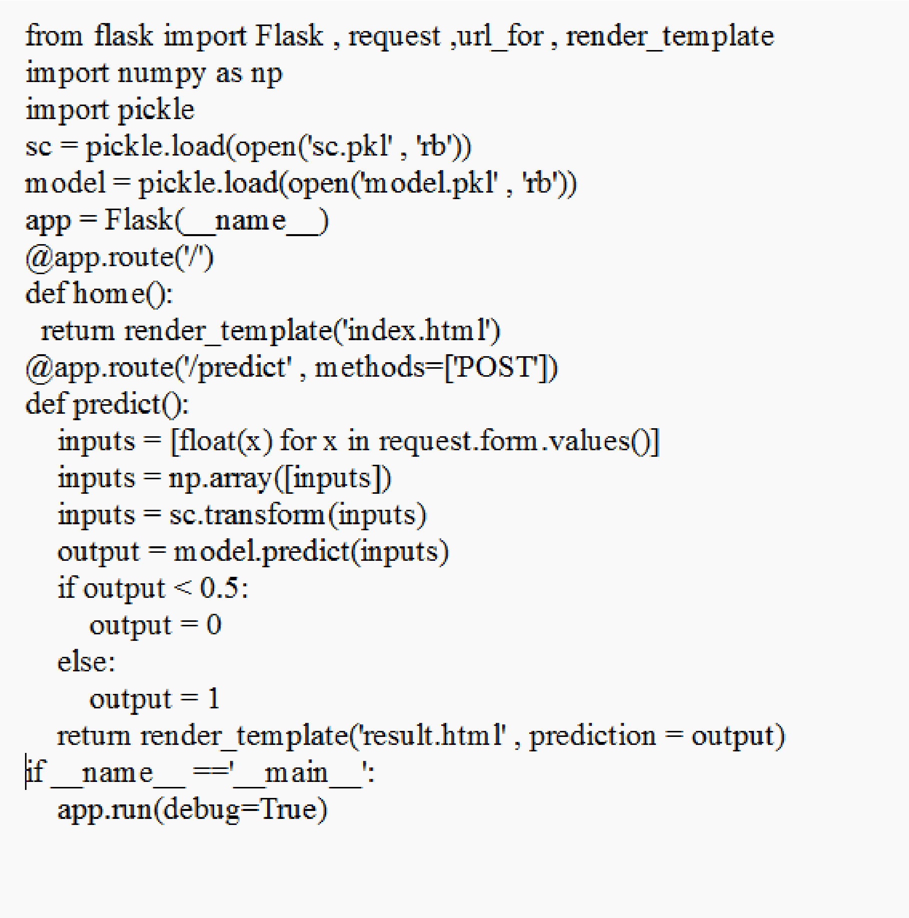


```
from sklearn.metrics import accuracy_score, precision_score
print(accuracy_score(y_test,knn_y_pred))
print(precision_score(y_test , knn_y_pred))
from sklearn.svm import SVC
svm_classifier = SVC(kernel = 'rbf', random_state = 0)
svm_classifier.fit(x_train, y_train)
svm_y_pred = svm_classifier.predict(x_test)
from sklearn.metrics import confusion_matrix
svm_cm = confusion_matrix(y_test, svm_y_pred)
sns.heatmap(svm_cm , annot=True)
from sklearn.metrics import accuracy_score, precision_score
print(accuracy_score(y_test,svm_y_pred))
print(precision_score(y_test , svm_y_pred))
import keras
from keras.models import Sequential
from keras.layers import Dense,  Dropout
# Initialising the ANN
classifier = Sequential()
# Adding the input layer and the first hidden layer
classifier.add(Dense(units = 400 , activation='relu' , input_dim=10))
classifier.add(Dropout(rate=0.1))
# Adding Second hidden layer
classifier.add(Dense(units = 400 , activation='relu'))
classifier.add(Dropout(rate=0.1))
# Output layer
classifier.add(Dense(units = 1 , activation='sigmoid'))
classifier.compile(optimizer='adam' , loss='binary_crossentropy' , metrics=['accuracy'])
classifier.fit(x_train, y_train, batch_size=32 , epochs = 100)
ann_y_pred = classifier.predict(x_test)
ann_y_pred[0]
ann_y_pred = ann_y_pred >=0.5
from sklearn.metrics import confusion_matrix
ann_cm = confusion_matrix(y_test, ann_y_pred)
sns.heatmap(ann_cm , annot=True)
```



```
from sklearn.metrics import confusion_matrix
ann_cm = confusion_matrix(y_test, ann_y_pred)
sns.heatmap(ann_cm , annot=True)
from sklearn.metrics import accuracy_score, precision_score
print(accuracy_score(y_test,ann_y_pred))
print(precision_score(y_test , ann_y_pred))
import pickle
pickle.dump(svm_classifier, open('model.pkl', 'wb'))
pickle.dump(sc, open('sc.pkl', 'wb'))
```

GUI:

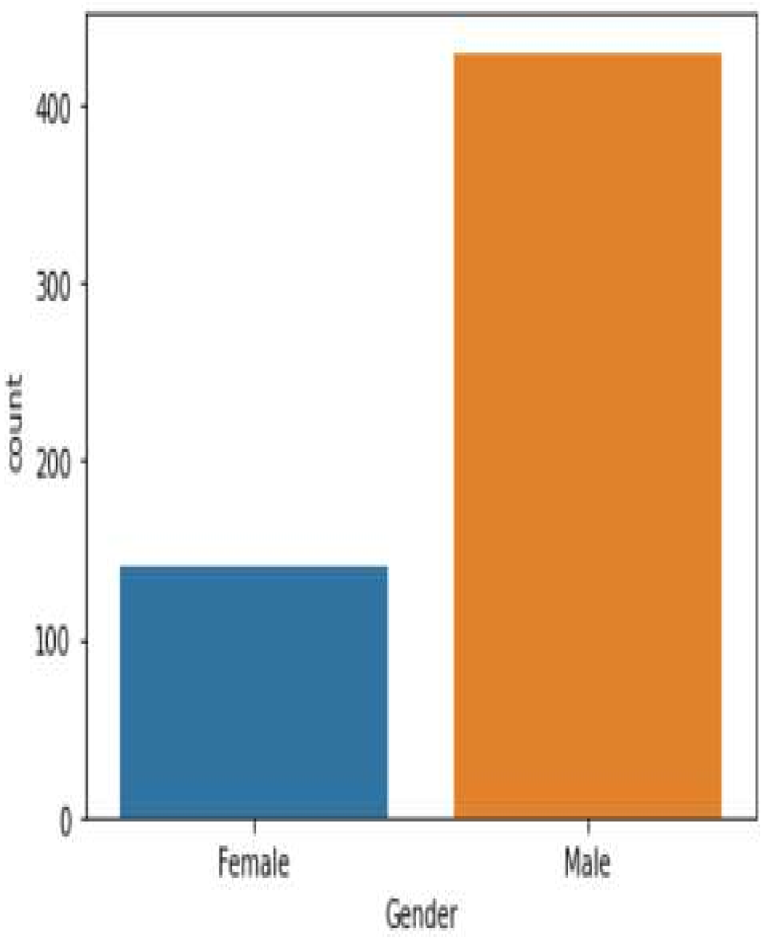


```
from flask import Flask , request ,url_for , render_template
import numpy as np
import pickle
sc = pickle.load(open('sc.pkl' , 'rb'))
model = pickle.load(open('model.pkl' , 'rb'))
app = Flask(__name__)
@app.route('/')
def home():
  return render_template('index.html')
@app.route('/predict' , methods=['POST'])
def predict():
    inputs = [float(x) for x in request.form.values()]
    inputs = np.array([inputs])
    inputs = sc.transform(inputs)
    output = model.predict(inputs)
    if output < 0.5:
        output = 0
    else:
        output = 1
    return render_template('result.html' , prediction = output)
if __name__ =='__main__':
    app.run(debug=True)
```

5. SCREENSHOTS

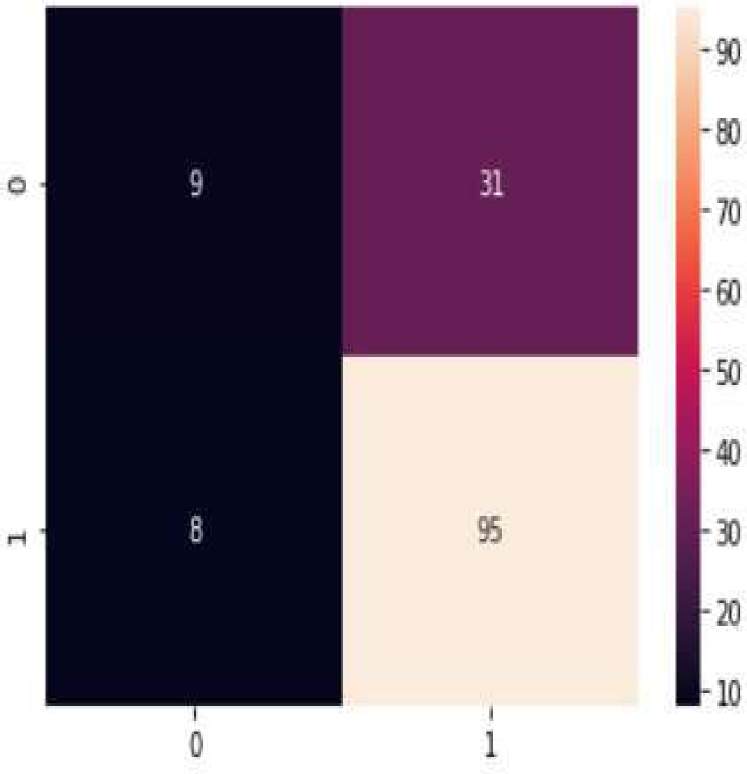
5. SCREENSHOTS

5.1 Bar Graph on count vs gender



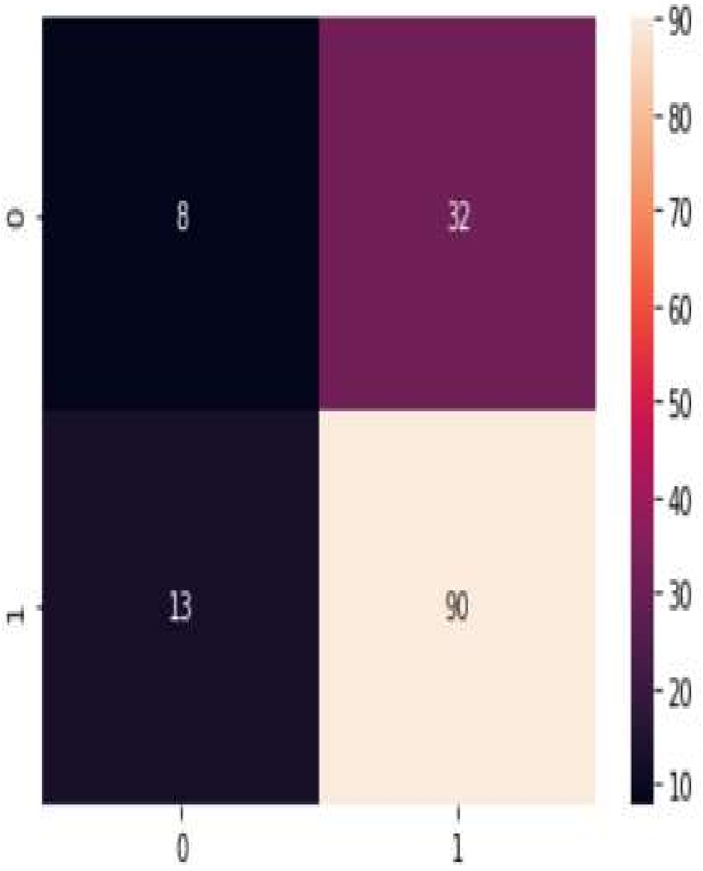
Screenshot 5.1:Bar Graph on count vs gender

5.2 Confusion Matrix of Logistic Regression



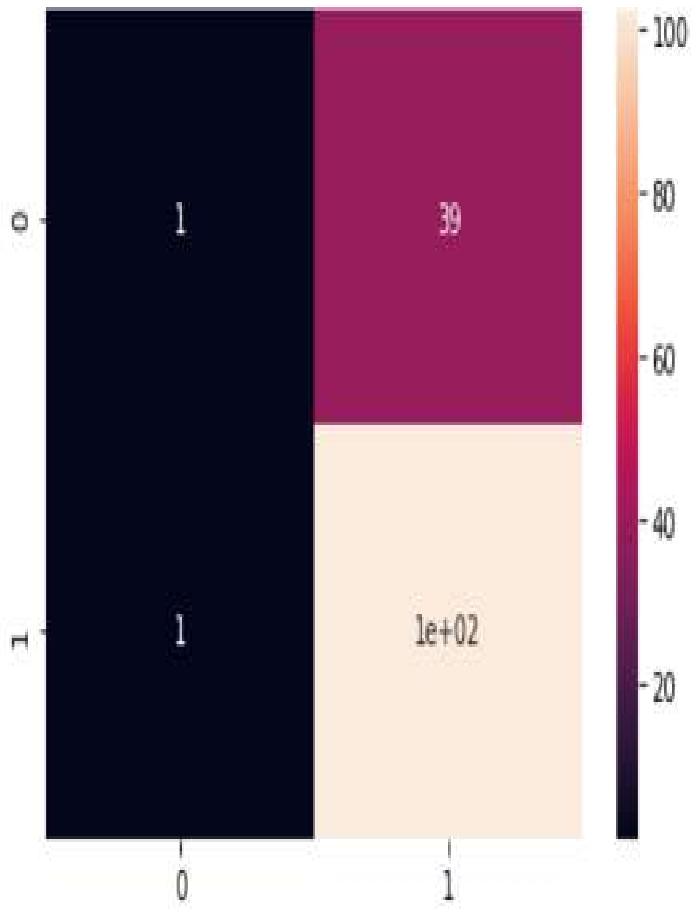
Screenshot 5.2: Confusion Matrix of Logistic Regression

5.4 Confusion Matrix of KNN



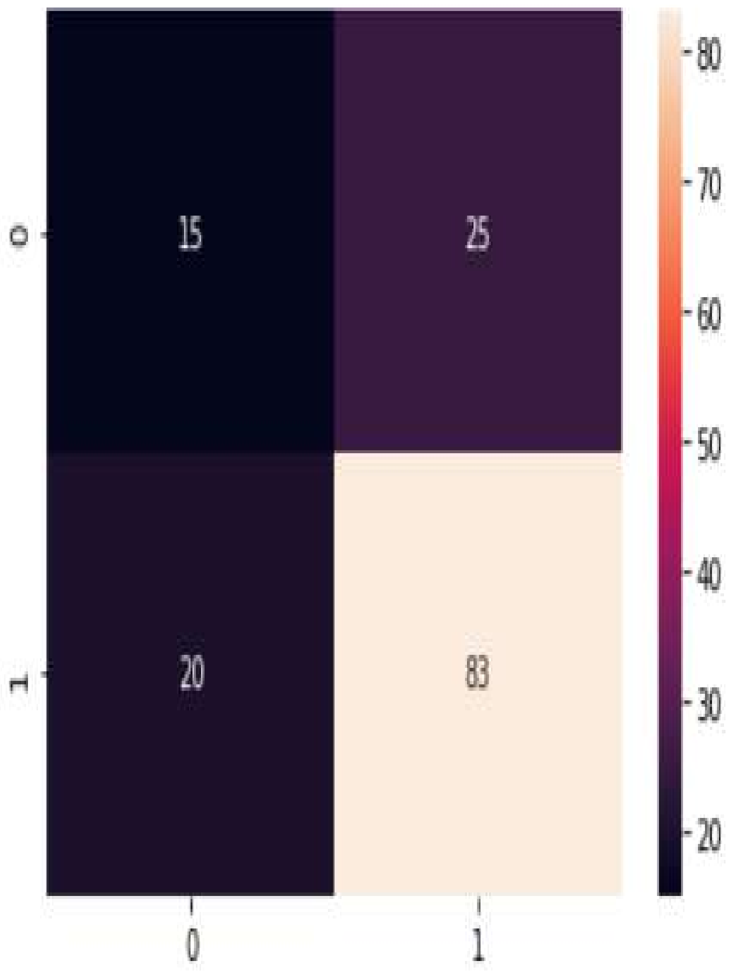
Screenshot 5.3: Confusion Matrix of KNN

5.1 Confusion Matrix of SVM



Screenshot 5.4: Confusion Matrix of SVM

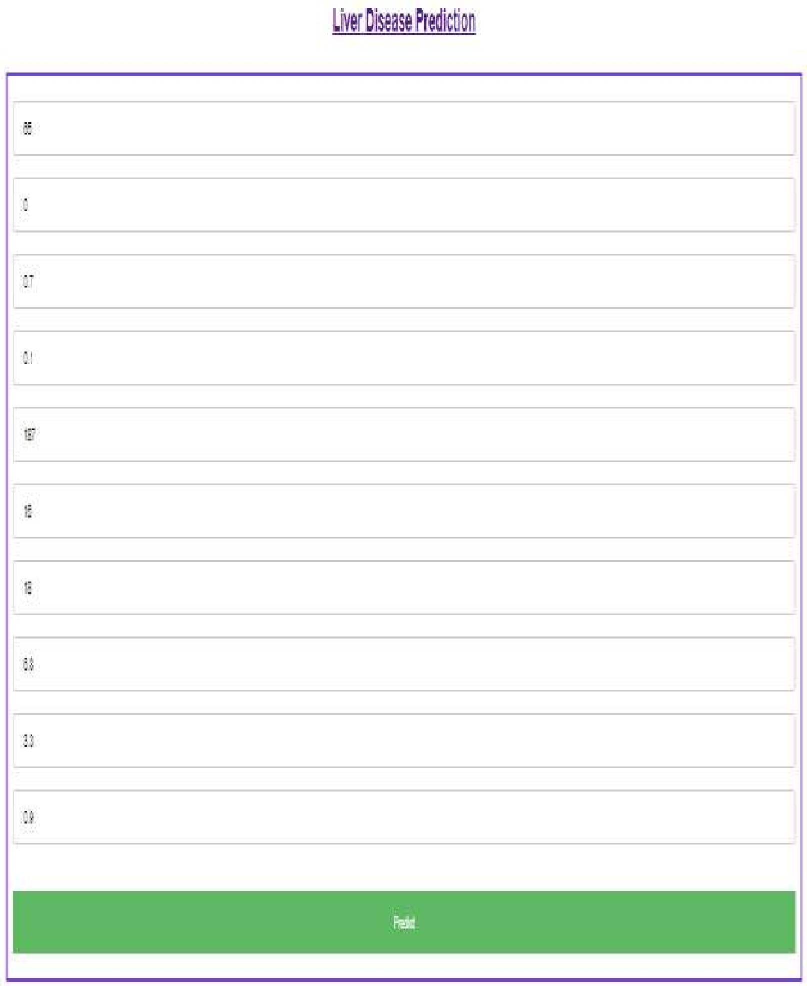
5.5 Confusion Matrix of ANN:



Screenshot 5.5: Confusion Matrix of ANN

5.2

5.3 RESULT:



Liver Disease Prediction

65

0

0.7

0.1

187

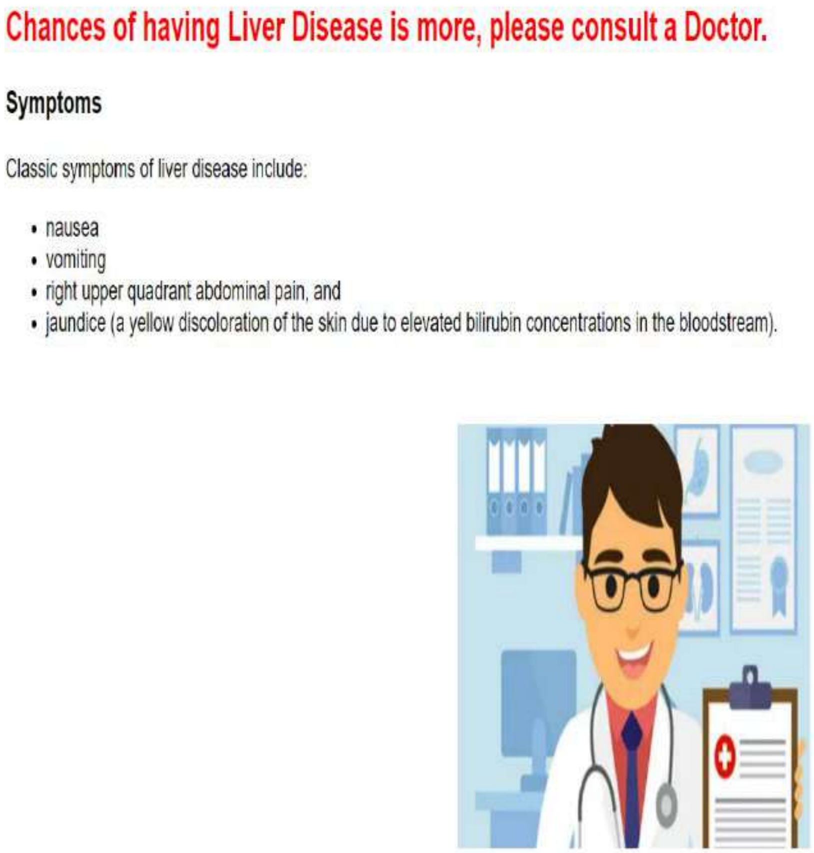
16

18

6.8

3.3

0.9



Chances of having Liver Disease is more, please consult a Doctor.

Symptoms

Classic symptoms of liver disease include:

- nausea
- vomiting
- right upper quadrant abdominal pain, and
- jaundice (a yellow discoloration of the skin due to elevated bilirubin concentrations in the bloodstream).

Screenshot 5.6 Result

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results. An individual component may be either an individual function or a procedure. Unit Testing is typically performed by the developer. In SDLC or V Model, Unit testing is the first level of testing done before integration testing. Unit testing is such a type of testing technique that is usually performed by developers. Although due to the reluctance of developers to test, quality assurance engineers also do unit testing.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components. In the real world, when applications are developed, it is broken down into smaller modules and individual developers are assigned 1 module. The logic implemented by one developer is quite different than another developer, so it becomes important to check whether the logic implemented by a developer is as per the expectations and rendering the correct value in accordance with the prescribed standards. Big bang approach integrates all the modules in one go i.e. it does not go for integrating the modules one by one. It verifies if the system works as expected or not once integrated. If any issue is detected in the completely integrated module, then it becomes difficult to find out which module has caused the issue. Big bang approach is a time-consuming process of finding a module which has a defect itself as that would take time and once the defect is detected, fixing the same would cost high as the defect is detected at the later stage.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.3 TEST CASES

6.3.1 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Classification test 1	To check if the classifier performs its task	Age: 65 Gender: Female Total Bilirubin: 0.7 Direct Bilirubin: 0.1 Alkaline Phosphatase: 187 Alamino Aminotransferase: 16 Aspartate Aminotransferase: 18 Total Proteins: 6.8 Albumin: 3.3 Albumin and Globulin Ratio: 0.9	1(YES)
2	Classification test 2	To check if the classifier performs its task	Age: 17 Gender: Male Total Bilirubin: 0.9 Direct Bilirubin: 0.3 Alkaline Phosphatase: 202 Alamino Aminotransferase: 22 Aspartate Aminotransferase: 19 Total Proteins: 7.4 Albumin: 4.1 Albumin and Globulin Ratio: 1.2	2(NO)

7. CONCLUSION

CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

In this project, we have proposed methods for diagnosing liver disease in patients using Machine learning techniques. The four machine learning techniques that were used include SVM , Logistic Regression , KNN and Artificial Neural Network. The system has been implemented using all the models and their performance was evaluated .The Performance evaluation was based on certain performance metrics. ANN was the model that resulted in highest accuracy with an accuracy of 98%. Comparing this work with the previous research works, it was discovered that ANN proved highly efficient. A GUI, which can be used as a medical tool by hospitals and medical staff was implemented using ANN.

7.2 PROJECT FUTURE SCOPE

The project has a very vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. The model can be improved by applying latest machine learning algorithms in the future. The future methodology is used to analyze the liver region into separable compartments i.e. liver etc. However, the method requires further improvement mostly regarding feature selection of the liver into multiple components: renal cortex, renal column, renal medulla and renal pelvis. Apart from that, it is planned to expand the database on which the system will be tested. And also the proposed method in this thesis can be employed for detecting the heart diseases in future with the heart dataset and classification of the diseases. Further, a lot of work is being done using multi-objective PSO for feature selection to improve liver classification performance and to reduce number of features selected as well. Most of the existing multi-objective feature selection based on PSO algorithms use binary tournament selection to select best and uniform and non-uniform mutation. There is a scope to further reduce search space for better liver classification accuracy if enhanced selection and mutation procedures are being used. Total_protiens, Albumin, A/G ratio, SGPT, SGOT, Alkphos to evaluate the performance of different classification algorithm. In future, we have attempted to classify different feature selection algorithms into four groups: complete search, heuristic search, meta-heuristic methods and methods that use artificial neural network(ANN).

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 GITHUB REPOSITORY LINKS

1. **https://github.com/Thanuja1810/LiverDisease.git**
2. https://github.com/shivani-madipeddi/Liver-Disease-Prediction.git
3. https://github.com/venkat-123456

8.2 REFERENCES

[1] Michael J Sorich. An intelligent model for liver disease diagnosis. Artificial Intelligence in Medicine 2009;47:53—62.

[2] Paul R. Harper, A review and comparison of classification algorithms for medical decision making.

[3] BUPA Liver Disorder Dataset. UCI repository machine learning databases.

[4] Prof Christopher N. New Automatic Diagnosis of Liver Status Using Bayesian Classification.

[5] Schiff's Diseases of the Liver, 10th Edition Copyright ©2007 Ramana, Eugene R.; Sorrell, Michael Maddrey, Willis C.

[6] P. Sug, On the optimality of the simple Bayesian classifier under zero-one loss, Machine Learning 29 (2–3) (1997) 103–130.

[7] L. Breiman, Random Forests. Machine Learning, 45(1), (2001); 5–32. https://doi.org/10.1023/A:1010933404324

[8] Decision Trees, Retrieve from: https://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/, Last Accessed: 5 Octobor,2019

[9] Support vector machine, Retrieve from: http://www.statsoft.com/textbook/support-vector-machines, Last Accessed: 5 Octobor,2019

[10] V. Vapnik, I. Guyon, T. H.-M, Learn, and undefined 1995. Support vector

machines. statweb.stanford.edu (1995).

[11] Zhang M, Zhou Z, "ML-KNN: A lazy learning approach to multi-label learning." Pattern recognition40.7:(2007); 2038-2048.

[12] G. Guo, H. Wang, D. Bell, Y. Bi, K. Greer, KNN Model Based Approach in Classification (pp. 986–996). Springer, Berlin, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62 .

[13] S. M. Mahmud, et al. "Machine Learning Based Unified Framework for Diabetes Prediction." Proceedings of the 2018 International Conference on Big Data Engineering and Technology. ACM (2018).

[14] S. Safavian, D. Landgrebe, A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3), (1991); 660-674.

[15] A. Chervonenkis, Early history of support vector machines. In Empirical Inference (pp. 13-20). Springer, Berlin, Heidelberg (2013).

8.3 WEBSITES

[1] www.accenture.com

[2] www.javatpoint.com

[3] www.edureka.co/blog/python-libraries/

[4] www.kaggle.com

[5] www.pantechelearning.com/

9. JOURNAL

e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

MACHINE LEARNING MODEL TO DETECT DISEASES IN LIVER

Dr. Punyaban Patel*[1], Thanuja Kommisetty*[2], Venkat Madipadigav*[3], Madipeddi Shivani*[4]

*[1]Professor, Department Of Computer Science And Engineering, CMR Technical Campus, Hyderabad, Telangana, India.

*[2,3,4]Student, Department Of Computer Science And Engineering, CMR Technical Campus, Hyderabad, Telangana, India.

ABSTRACT

During the recent decades, the risk of Liver disease in people is increasing at a rapid rate and is sought to be one of the fatal diseases in the world. It's quite a difficult task for researchers to predict the disease from humongous medical databases. To combat this issue, they have come up with machine learning techniques like classification and clustering. The main aim of this project is to predict the chances of a patient having a liver disease using the classification algorithms. These algorithms are compared based on their classification accuracy and execution time. With these performance factors taken into consideration, the algorithm which serves as a better classifier is chosen. Finally, a GUI (Graphical User Interface), used as a medical tool by hospitals is implemented in accordance with the best classifier.

Keywords: Logistic Regression, Neural Network, Dataset, Accuracy, SVM.

I. INTRODUCTION

This project provides the software which facilitates to upload the details and get to know the prediction for Liverdisease. This project uses Machine Learning algorithms to classify whether the liver condition is normal. We use KNN, ANN, SVM and Logistic Regression models for the prediction. This model will be useful for health industries who need to predict the diseases. The model will be helpful to know whether the liver condition is normal or abnormal using the blood reports of the patient. This information regarding the patients will be helpful for the medical companies in the process. The existing models include various machine learning techniques which yield outputs of less accuracy and can't handle large bundles of data. The poor performance in the training and testing of the liver datasets is observed. These previously designed systems have been sufficient but more work has to be done on their prediction rate for better accuracy in the diagnosis of the liver disease. The proposed system here uses concept of machine learning, and the models are first trained, then tested. Finally the most accurate model will predict the final result. Initially, the system asks you to enter your details including age, gender, total Bilirubin, direct Bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT and Alkphos. These values can be known by blood test report of the user. After taking these inputs from the user, the system compares the data input with the training dataset of most accurate model and then predicts the result accordingly as risk or no risk. The algorithms used are Logistic Regression, K-Nearest Neighbor (KNN), Artificial Neural Network (ANN) and Support Vector Machine (SVM). The dataset used is The Indian Liver Patient Dataset (ILPD) which was selected from UCI Machine learning repository. It is a sample of the entire Indian population collected from Andhra Pradesh region and comprises of 585 patients data. The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations.

II. METHODOLOGY

The various stages involved are:

Exploratory Data Analysis

Data visualization: With the help of data visualization, we can see how the data looks like and what kind of correlation is held by the attributes of data. It is the fastest way to see if the features correspond to the output features.

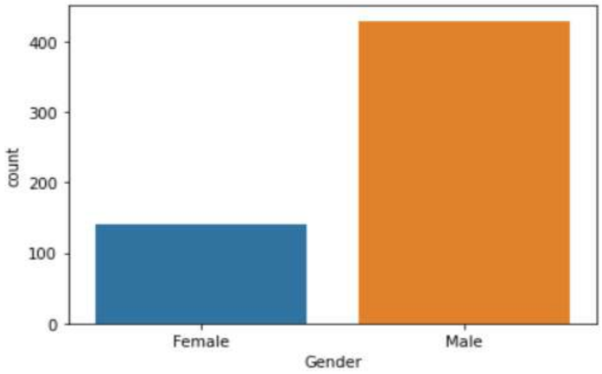


Figure 1: Bar Graph on Gender vs count

Correlation Analysis: Correlations have three important characteristics. They can tell us about the direction of the relationship, the form (shape) of the relationship, and the degree (strength) of the relationship between two variables**.**

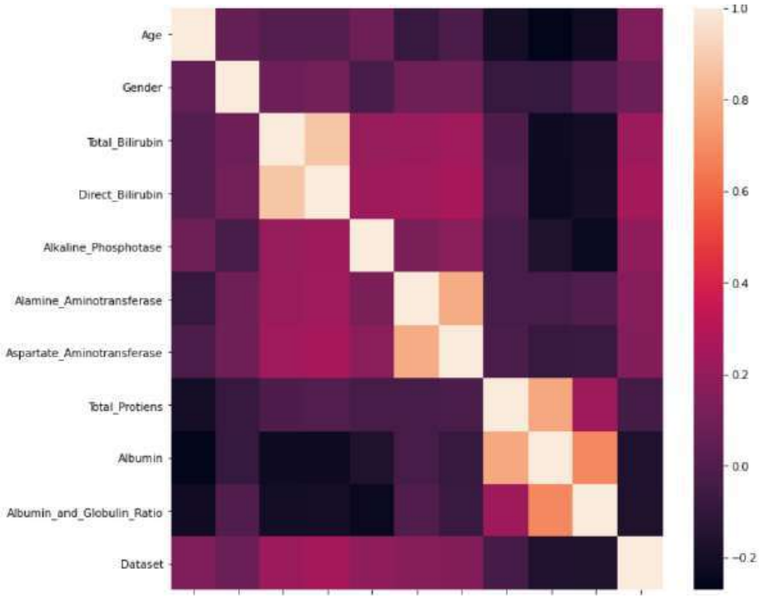


Figure 2: Correlation Matrix of the Model

Data Preprocessing

This involves eliminating the null and most common words from the text. The words in the dataset consists of links, multiple full stops, very long and short words. These all need to be eliminated before providing it to the algorithm. The significant stages in data preprocessing are Data Cleaning, Data Integration, Data Reduction and Data Tranformation. It is carried out to meet the criteria of accuracy, completeness, consistency, timeliness, believability and interpretability.

Feature Engineering

Label Encoder : Provided by Sklearn library that converts the levels of categorical features (labels) into numeric form so as to convert it into the machine readable form. It encode labels with a value between 0 and n_classes-1 where n is the number of distinct labels. If a label repeats it assigns the same value to as assigned earlier. We choose label encoding over one-hot encoding to reduce the pre-processing time, and majority due to the fact that there are predefined values under gender and smoker. It seems like a better option to the curse of dimensionality in the feature space. Count Vectorizer is used to convert a collection of text documents to a vector of term/token counts and build a vocabulary of known words, but also to encode new documents using that vocabulary. It also enables the pre-processing of text data prior to generating the vector representation. Here, we use stop_words='english' with Count Vectorizer since this just counts the occurrences of each word in its vocabulary, extremely common words like 'the', 'and', etc. will become very important features while they add little meaning to the text. This is an important step in pre-processing.

Training Classification Model

We split the dataset into testing and training in multiple ratios to give the best results. Now we train the model using the Machine Learning algorithms namely: Logistic Regression, KNN, ANN and SVM to predict the exact result.

III. MODELING AND ANALYSIS

The Block diagram of the Model is as follows:

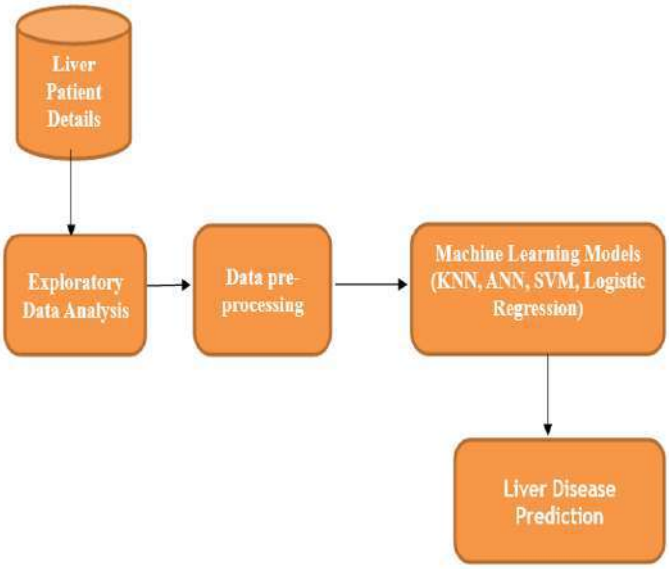

Liver Patient Details
Exploratory Data Analysis
Data pre-processing
Machine Learning Models (KNN, ANN, SVM, Logistic Regression)
Liver Disease Prediction

Figure 3: Block Diagram of the Model

The various Machine Learning Models used are:

LOGISTIC REGRESSION:

Logistic regression is one of the simpler classification models. Because of its parametric nature it can to some extent be interpreted by looking at the parameters making it useful when experimenters want to look at relationships between variables. The name logistic regression is a bit unfortunate since a regression model is usually used to find a continuous response variable, whereas in classification the response variable is discrete. The term can be motivated by the fact that we in logistic regression found the probability of the response variable belonging to a certain class. The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE). Once the optimal coefficient (or coefficients if there is more than one independent variable) is found, the conditional probabilities for each observation can be calculated, logged, and summed together to yield a predicted probability. For binary classification, a probability less than .5 will predict 0 while a probability greater than 0 will predict 1. After the model has been computed, it's best practice to evaluate the how well the model predicts the dependent variable, which is called goodness of fit.

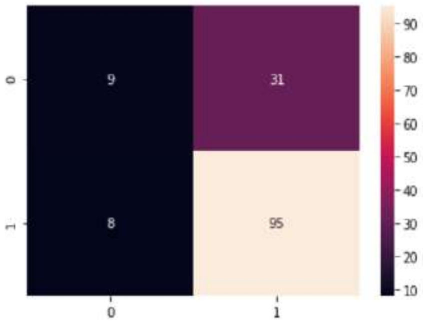


Figure 4: Confusion Matrix Of Logistic Regression

K-NEAREST NEIGHBOUR:

KNN This section describes the implementation details of KNN algorithm. The model for K-Nearest Neighbor is the entire training dataset. When a prediction is required for a unseen data instance, the KNN algorithm will search through the training dataset for the k-most similar instances. For classification problems, a class label is assigned on the basis of a majority vote-i.e. the label that is most frequently represented around a given data point is used. While this is technically considered "plurality voting", the term, "majority vote" is more commonly used in literature. The distinction between these terminologies is that "majority voting" technically requires a majority of greater than 50%, which primarily works when there are only two categories. When you have multiple classes-e.g. four categories, you don't necessarily need 50% of the vote to make a conclusion about a class; you could assign a class label with a vote of greater than 25%.

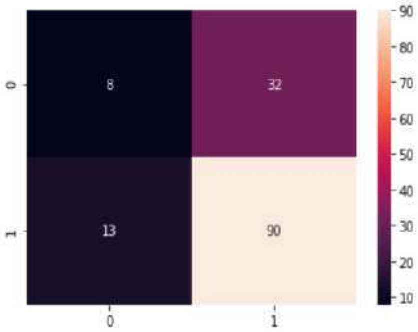


Figure 5: Confusion Matrix Of KNN

ARTIFICIAL NEURAL NETWORK:

A back propagation neural network was designed. In this network, 10 input neurons were present at the input layer. The number of inputs represents the total number of attributes in the dataset. The input layer uses Rectified Linear Unit activation function. The output layer contains a single layer which uses the sigmoid activation function. In order to obtain a required recognition rate that is capable enough to diagnose the liver disorder in a patient. There is a need for varying certain parameters in the neural network models to produce the required optimum result. These parameters are the learning rate, momentum rate and the hidden neurons. All these parameters present in the backpropagation neural networks. The learning rate is the learning power of the system, the momentum rate determines the learning speed of the system. The number of hidden neurons in the network has to be varied to produce the optimal result. The numbers of neurons needed at the hidden layer are experimenting in order to deter-mine the best neurons that can represent the features present in the input dataset accurately to produce the required optimum result. The numbers of neurons required in the hidden layer were experimenting by varying the neurons. The sigmoid function was used in the output layer because of its soft switching ability and simplicity in derivatives.

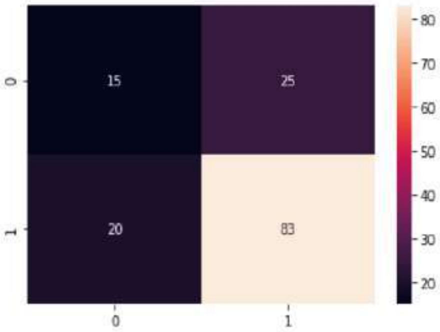


Figure 6: Confusion Matrix Of ANN

SUPPORT VECTOR MACHINE:

SVM aims to find an optimal hyperplane that separates the data into different classes. The scikit-learn package in python is used for implementing SVM. The pre-processed data is split into test data and training set which is of 25% and 75% of the total dataset respectively. A support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space. A good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds.

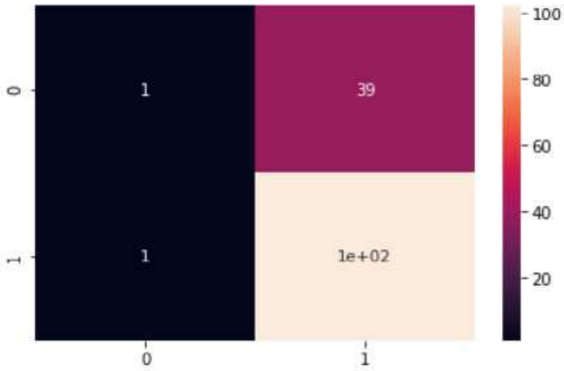


Figure 7: Confusion Matrix Of SVM

IV. RESULTS AND DISCUSSION

Our main goal going into this project was to predict liver disease using various machine learning techniques. We predicted using Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbor (K-NN) and Neural Network. All of them predicted with better results. With Each algorithm, we have observed Accuracy, Precision, Sensitivity and Specificity as follows:

Table 1. Comparison of performance of all 4 cases

Classification Algorithm	Accuracy	Precision	Sensitivity	Specificity
Logistic Regression	73.23	78.57	88	30.62
K-NN	72.05	80.98	83.78	44.04
SVM	75.04	77.09	79	71.11
ANN	92.8	93.78	97.23	83

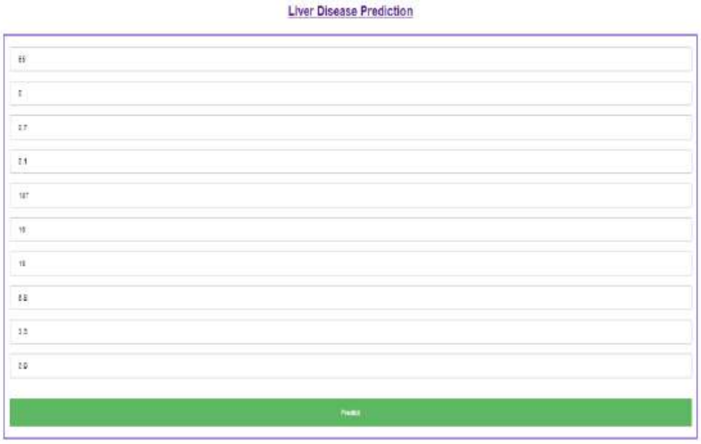
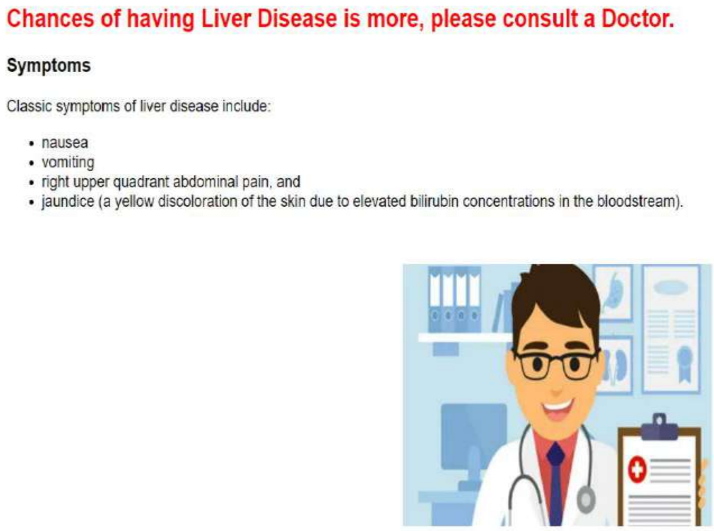


Figure 8: Output in GUI



Chances of having Liver Disease is more, please consult a Doctor.

Symptoms

Classic symptoms of liver disease include:

- nausea
- vomiting
- right upper quadrant abdominal pain, and
- jaundice (a yellow discoloration of the skin due to elevated bilirubin concentrations in the bloodstream).

Figure 9: Output in GUI

V. CONCLUSION

In this project, we have proposed methods for diagnosing liver disease in patients using Machine learning techniques. The four machine learning techniques that were used include SVM , Logistic Regression , KNN and Artificial Neural Network. The system has been implemented using all the models and their performance was evaluated .The Performance evaluation was based on certain performance metrics. ANN was the model that resulted in highest accuracy with an accuracy of 98%.

ACKNOWLEDGEMENT

We thank CMR Technical Campus for supporting this paper titled with "MACHINE LEARNING MODEL TO DETECT DISEASES IN LIVER" ,which provided good facilities and support to accomplish our work. Sincerely thanks to our Chairman, Director, Deans, Guide and faculty members for giving valuable suggestions and guidance in every aspect of our work.

VI. REFERENCES

[1] Paul R. Harper, A review and comparison of classification algorithms for medical decision making.

[2] BUPA Liver Disorder Dataset. UCI repository machine learning databases.

[3] Prof Christopher N. New Automatic Diagnosis of Liver Status Using Bayesian Classification.

[4] Schiff's Diseases of the Liver, 10th Edition Copyright ©2007 Ramana, Eugene R.; Sorrell, Michael Maddrey, Willis C.

[5] P. Sug, On the optimality of the simple Bayesian classifier under zero-one loss, Machine Learning 29 (2–3) (1997) 103–130.

[6] 16th Edition HARRISON'S PRINCIPLES of Internal Medicine.

[7] Michael J Sorich. An intelligent model for liver disease diagnosis. Artificial Intelligence in Medicine 2009;47:53—62.

IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 06/40600163655

Date: 25/06/2022

Certificate of Publication

*This is to certify that author "**Thanuja Kommisetty**" with paper ID "**IRJMETS40600163655**" has published a paper entitled "MACHINE LEARNING MODEL TO DETECT DISEASES IN LIVER" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 06, June 2022***

IRJMETS
Impact Factor
6.752

International Research Journal of Modernization in Engineering Technology and Science

Editor in Chief

We Wish For Your Better Future
www.irjmets.com

IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 06/40600163655

Date: 25/06/2022

Certificate of Publication

*This is to certify that author "**Venkat Madipadigav**" with paper ID "**IRJMETS40600163655**" has published a paper entitled "MACHINE LEARNING MODEL TO DETECT DISEASES IN LIVER" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 06, June 2022***

Editor in Chief

IRJMETS
Impact Factor
6.752

International Research Journal of Modernization in Engineering Technology and Science

We Wish For Your Better Future

www.irjmets.com

IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 4/Issue 06/40600163655

Date: 25/06/2022

Certificate of Publication

*This is to certify that author "**Madipeddi Shivani**" with paper ID "**IRJMETS40600163655**" has published a paper entitled "MACHINE LEARNING MODEL TO DETECT DISEASES IN LIVER" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 4, Issue 06, June 2022***

IRJMETS
Impact Factor
6.752

International Research Journal of Modernization in Engineering Technology and Science

Editor in Chief

We Wish For Your Better Future
www.irjmets.com