**Liver Disease Prediction**

A MINI PROJECT REPORT SUBMITTED BY

Samit D Manvar S Dhruva

4NM18CS152 4NM18CS140

VI Semester ,C Section VI Semester ,C Section

UNDER THE GUIDANCE OF

Dr Sarika Hegde

Associate Professor

Department of Computer Science and Engineering

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

Bachelor of Engineering in Computer Science & Engineering

from

Visvesvaraya Technological University, Belagavi



N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum)

AICTE approved, (ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

NITTE -574 110, Udupi District, KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

Date Of Submission : 31-05-2021



(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

🕿: 08258 - 281039 – 281263, Fax: 08258 – 281265

### Department of Computer Science and Engineering

B.E.CSE Program Accredited by NBA,New Delhi from 1-7-2018 to 30-6-2021

CERTIFICATE

“Prediction Of Liver Disease” is a bonafide work carried out by Samit D Manvar (4NM18CS152) and S Dhruva (4NM18CS40) in partial fulfilment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2020-2021.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

**Signature of Guide Signature of HOD**

**ACKNOWLEDGEMENT**

We believe that our project will be complete only after we thank the people who have contributed to make this project successful. First and foremost, our sincere thanks to our beloved principal, Dr. Niranjan N. Chiplunkar for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities. We sincerely thank Dr.Jyothi Shetty, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte. We express our deep sense of gratitude and indebtedness to our guide Dr Sarika Hegde, Associate Professor , Department of Computer Science and Engineering, for her inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We thank all the teaching and non-teaching staff members of the Computer Science and Engineering Department and our parents and friends for their honest opinions and suggestions throughout the course of our project. Finally, we thank all those who have supported us directly or indirectly throughout the project and making it a grand success.

Samit D Manvar(4NM18CS152)

S Dhruva(4NM18CS140)

**ABSTRACT**

Today’s Healthcare is very important aspect for every human ,so there is a need to provide medical service that are easily available to everyone. In this mini project ,the main focus is to predict whether a person is suffering from liver disease or not with help of machine learning model (approach).

Liver disease is any disturbance of [liver function](https://www.medicinenet.com/liver_anatomy_and_function/article.htm) that causes illness. The liver is responsible for many critical functions within the body and should it become diseased or injured, the loss of those functions can cause significant damage to the body. Liver disease is also referred to as hepatic disease.

The implementation is done on Indian Liver Patient Dataset(ILPD) from north east of Andhra Pradesh ,India .The different attributes such as Age,Gender,Total Bilirubin,Direct Bilrubin,Alkaline Phosphate,  Alamine Aminotransferase, Aspartate Aminotransferase, Total Protiens, Albumin, Albumin and Globulin Ratio,Selector field used to split the data into two sets (labeled by the experts).

The two classification algorithms - Logistic Regression and Decision Tree are implemented on the dataset to find the accuracy.

# **Table Of Contents**

**Cover Page………………………………………………………………………………………1**

**Certificate Page………………………………………………………………………………...2**

**Acknowledgement……………………………………………………………………………..3**

**Abstract………………………………………………………………………………………….4**

[**Table of Contents………………………………………………………………………………5**](#_30j0zll)

**Introduction……………………………………………………………………………………..6**

**Literature Survey……………………………………………………………………………….7**

**Design…………………………………………………………………………………………….8**

**Implementation………………………………………………………………………………...10**

**Results…………………………………………………………………………………………..18**

**Conclusion……………………………………………………………………………………...20**

**References……..……………………………………………………………………………….20**

**Snapshots of the Project …...……………………………………………………………….21**

**Introduction**

**Machine** **Learning** is a subset of AI that uses data to solve tasks. These solvers are trained models of data that learn based on the information provided to them. ML algorithms use our data to learn and automatically solve predictive tasks. Machine learning is a buzzword for today's technology, and it is growing very rapidly day by day. We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Some of the trending real-world applications of Machine Learning include speech recognition, disease diagnosis, online fraud detection etc. In our mini project we are going to implement one of the application of machine learning -prediction of liver disease.

The liver is essential for digesting food and ridding your body of toxic substances.Liver disease can be inherited (genetic). Liver problems can also be caused by a variety of factors that damage the liver, such as viruses, alcohol use and obesity.

1. A patient going to a doctor with certain symptoms.

2. The doctor recommending certain tests like blood test, urine test etc depending on the symptoms.

3. The patient taking the aforementioned tests in an analysis lab.

4. The patient taking the reports back to the hospital, where they are examined and the disease is identified.

The aim of this mini project is to somewhat reduce the time delay caused due to the unnecessary back and forth shuttling between the hospital and the pathology lab.A machine learning algorithm will be trained to predict whether a person is suffering from liver disease or not.

**Literature Survey**

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output. In supervised learning, models are trained using labelled dataset, where the model learns about each type of data.

Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output. Initially data pre-processing is carried out. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

EDA(Exploratory data analysis) is an approach of [analyzing](https://en.wikipedia.org/wiki/Data_analysis) [data sets](https://en.wikipedia.org/wiki/Data_set) to summarize their main characteristics, often using [statistical graphics](https://en.wikipedia.org/wiki/Statistical_graphics) and other [data visualization](https://en.wikipedia.org/wiki/Data_visualization) methods and also helps us to draw inference from the data set .

Under Supervised learning we can further categorize into regression and classification. Since our dataset has fixed number of features and one target column(output variable) which is categorical , our mini project is a classification based supervised learning. The two classification algorithms we have used is decision tree and logistic regression.

**Design**



Preparing the data









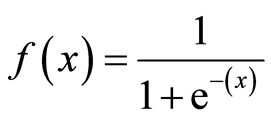
The above diagram shows the overall design of the model to predict the liver disease.We start creating the machine learning model by

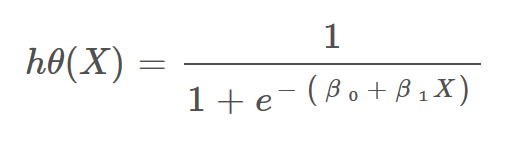
1. **Collecting the dataset :-** Here we have taken the indian liver disease dataset. This step is very crucial as the quality and quantity of data gathered will directly determine how good the predictive model will turn out to be.
2. **Preparing the data:-** After the training data is gathered, you move on to the next step of machine learning: Data preparation, where the data is loaded into a suitable place and then prepared for use in machine learning training. Here, the data is first put all together and then the order is randomized as the order of data should not affect what is learnt .It also includes cleaning the data such as filling or removing the null values present .Here in our dataset we have only 4 null values present in Albumin\_and\_Globulin\_Ratio . Apart from cleaning we also do necessary encoding (Label Encoder) based upon the type of algorithm chosen .We have encoded the Gender and the Target Dataset column to be encoded particularly for logistic regression model. This is also a good enough time to do any visualizations of the data, as that will help you see if there are any relevant relationships between the different variables, how you can take their advantage and as well as show you if there are any data imbalances present.
3. **Choosing the Right Algorithm:**-Since our model is a supervised classification based model we have chosen logistic regression and decision tree algorithm.
4. **Training and Testing data:-** Data needs to be split into two parts. The first part that is used in training our model, will be the majority of the dataset and the second will be used for the evaluation of the trained model’s performance. The other forms of adjusting and manipulation like normalization, error correction, and more take place at this step.In our mini project we have taken 75% of data for training and 25% of data for testing.
5. **Compare the results of Classifiers:-** Find out which model gives better accuracy.
6. **Performance Evaluation:-**Trying to improve the accuracy of model by feature selection or parameter tuning etc.
7. **Development of web application:-** Deploying the model with help of different tools , API , etc for easy interaction with the users.
8. **Predict:**-Finally the patients can check if they are suffering from liver disease or not by entering their blood report values and get the results.

**IMPLEMENTATION**

**LOGISTIC REGRESSION**

Logistic Regression uses a more complex cost function, this cost function can be defined as the **Sigmoid function**.Sigmoid function is used to map predicted values to probabilities.We use the Sigmoid function.





# **Decision Boundary**

We expect our classifier to give us a set of outputs or classes based on probability when we pass the inputs through a prediction function and returns a probability score between 0 and 1.

**Cost Function:-**The cost function gives you the measure of how far the predicted output(calculated hypothesis ‘h’) is from the original output.The cost function for logistic regression is

# 

# **Here y is actual output and h is predicted output.**

import numpy as np #importing necessary libraries

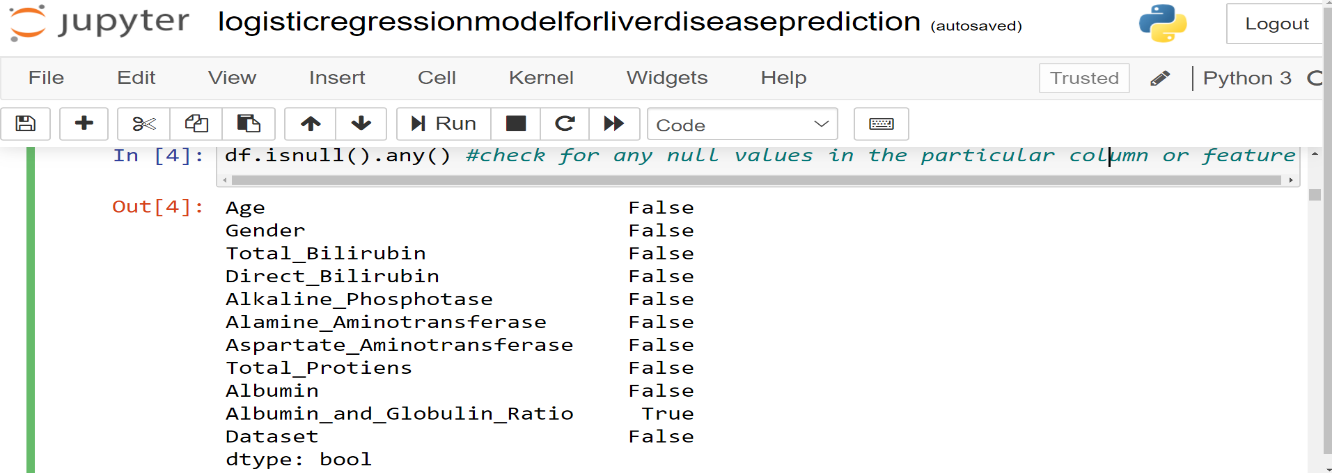
import pandas as pd

import seaborn as sns

df=pd.read\_csv('D:\\Downloads\\indian\_liver\_patient.csv') #load the dataset

df.head(5)

df.isnull().any() #check for any null values in the particular column or feature



#Albumin\_and\_Globulin\_Ratio has null values

df.isnull().sum() #4 null values in column Albumin\_and\_Globulin\_Ratio

df["Albumin\_and\_Globulin\_Ratio"].fillna(df['Albumin\_and\_Globulin\_Ratio'].mean(), inplace = True)

#fill null values by mean of that particular column

X = df.iloc[:,:-1].values #features

Y = df.iloc[:,-1].values #target

for u in range(len(Y)):

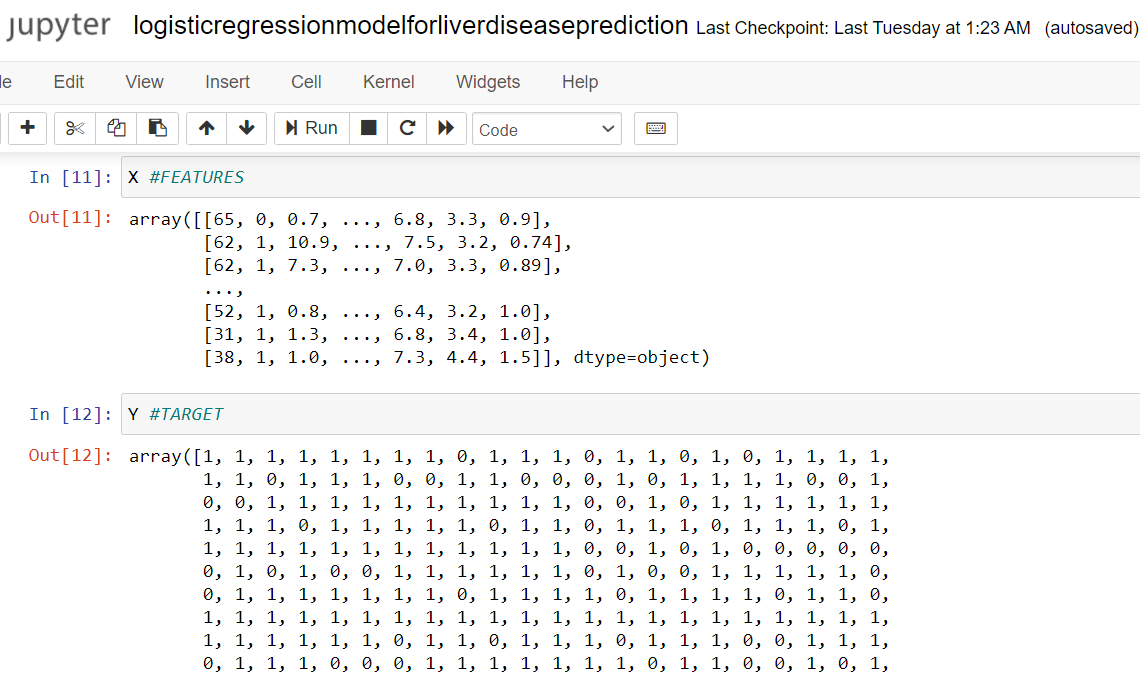
if Y[u] == 2: #converting all 2’s to 0’s for target column

Y[u] = 0 # encoding 1: liver disease, 0: no liver disease

from sklearn.preprocessing import LabelEncoder

lbl = LabelEncoder()

X[:,1] = lbl.fit\_transform(X[:,1]) #female : 0, male :1



sns.countplot(data=df, x = 'Dataset', label='Count')

l,nl = df['Dataset'].value\_counts()

print('Number of patients diagnosed with liver disease: ',l)

print('Number of patients not diagnosed with liver disease: ',nl)

Number of patients diagnosed with liver disease: 416

Number of patients not diagnosed with liver disease: 167



import seaborn as sns

sns.countplot(data=df, x = 'Dataset', label='Count')

mp,fp = df['Gender'].value\_counts()

print('no of male patients: ',mp)

print('Number of female patients: ',fp)

#0 female 1 male No of male patients: 441 Number of female patients: 142



from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test=train\_test\_split(X,Y,random\_state=18,test\_size=0.25)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

p=X\_train

q=Y\_train #p and used for logistic regrerssion without scikitklearn

#sigmod function which returns the probablity

#loss() function (log-loss) to reduce the cost (J)

#grad() function to update the weights

def sigmoid(x,w,b):

return 1/(1+np.exp(-(np.dot(x,w)+b)))

def loss(x,w,y,b):

s=sigmoid(x,w,b) #s is calculated output y is original output

return np.mean(-(y\*np.log(s))- ((1-y)\*np.log(1-s)))

def grad(x,y,w,b):

s=sigmoid(x,w,b)

return np.dot(x.T,(s-y))/p.shape[0] #p= size of feature dataset

#Accuracy() to find the accuracy of model

#gradientdescent() that performs the updation of weights , bias and loss(cost)

def accuracy(y\_pred,y\_test):

return np.mean(y\_pred==y\_test)

def gradientdescent(x,y):

w=np.zeros((x.shape[1]))

b=np.zeros(1)

ite=2000 #number of iteration to update weights

eta=0.1 #learning rate or say it as alpha

loss\_v=[] #loss

for i in range(ite):

probability=sigmoid(x,w,b)

l=loss(x,w,y,b)

gradient=grad(x,y,w,b)

w=w-(eta\*gradient) #update the weights

b=b-(eta\*np.sum(probability-y)/x.shape[0]) #update bias

loss\_v.append(l) #used to plot graph of loss (log loss)

print(l)

return w,b,loss\_v #returns the updated weights ,bias and loss(J)

w,b,loss\_v=gradientdescent(p,q) #w weight b bias loss\_v is log loss

# P AND Q ARE X\_TRAIN AND Y\_TRAIN

y\_pred=sigmoid(X\_test,w,b) #sigmoid function which calculates the probablity

for j,i in enumerate(y\_pred):

if i<0.5: #if value is less than 0.5 it belongs to class 0

y\_pred[j]=0

else: #else it belongs to class 0

y\_pred[j]=1

print('test accuracy boy',accuracy(y\_pred,Y\_test)\*100)

#without using buit in function the accuracy is around 78.08%

from sklearn.linear\_model import LogisticRegression #comparing with sklearn

logreg = LogisticRegression()

logreg.fit(X\_train, Y\_train)

log\_predicted= logreg.predict(X\_test)

from sklearn.metrics import accuracy\_score

print('Accuracy: \n', accuracy\_score(Y\_test,log\_predicted)\*100)

#Using built in function accuracy is 78.76

**DECISION TREE**

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree.

* **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
* **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
* **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
* **Step-4:** Generate the decision tree node, which contains the best attribute.
* **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

**Attribute selection measure or ASM used is Information Gain.**

Information Gain

* Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
* It calculates how much information a feature provides us about a class.

Information Gain= Entropy(S)- [(Weighted Avg) \*Entropy(each feature)

**Entropy:** Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

* S= Total number of samples
* P(yes)= probability of yes
* P(no)= probability of no

**IMPLEMENTATION OF DECISION TREE**

1.Data preprocessing step.

2. Fitting a Decision-Tree algorithm to the Training set

classifier= DecisionTreeClassifier(criterion="entropy")

classifier.fit(x\_train, y\_train)

3. Predicting the test result

y\_pred = classifier.predict(x\_test)

4. Test accuracy of the result

#Calculating the accuracy

x=0.0000000000

for i in range(0,30)

classifier= DecisionTreeClassifier(criterion="entropy")

classifier.fit(x\_train, y\_train)

y\_pred = classifier.predict(x\_test)

x=x+metrics.accuracy\_score(y\_test, y\_pred)

print("Accuracy:",i," :",metrics.accuracy\_score(y\_test, y\_pred))

print("Average accuracy of 30 runs:",x/30)

5.Visualising the decision tree by plotting a graph using graphviz

from sklearn.tree import export\_graphviz

from sklearn.externals.six import StringIO

from IPython.display import Image

col\_names=['Age', 'Gender', 'Total\_Bilirubin', 'Direct\_Bilirubin', 'Alkaline\_Phosphotase', 'Alamine\_Aminotransferase', 'Aspartate\_Aminotransferase','Total\_Protiens', 'Albumin', 'Albumin\_and\_Globulin\_Ratio']

import os

os.environ['PATH'] = os.environ['PATH']+';'+os.environ['CONDA\_PREFIX']+r"\Library\bin\graphviz"

import pydotplus

dot\_data = StringIO()

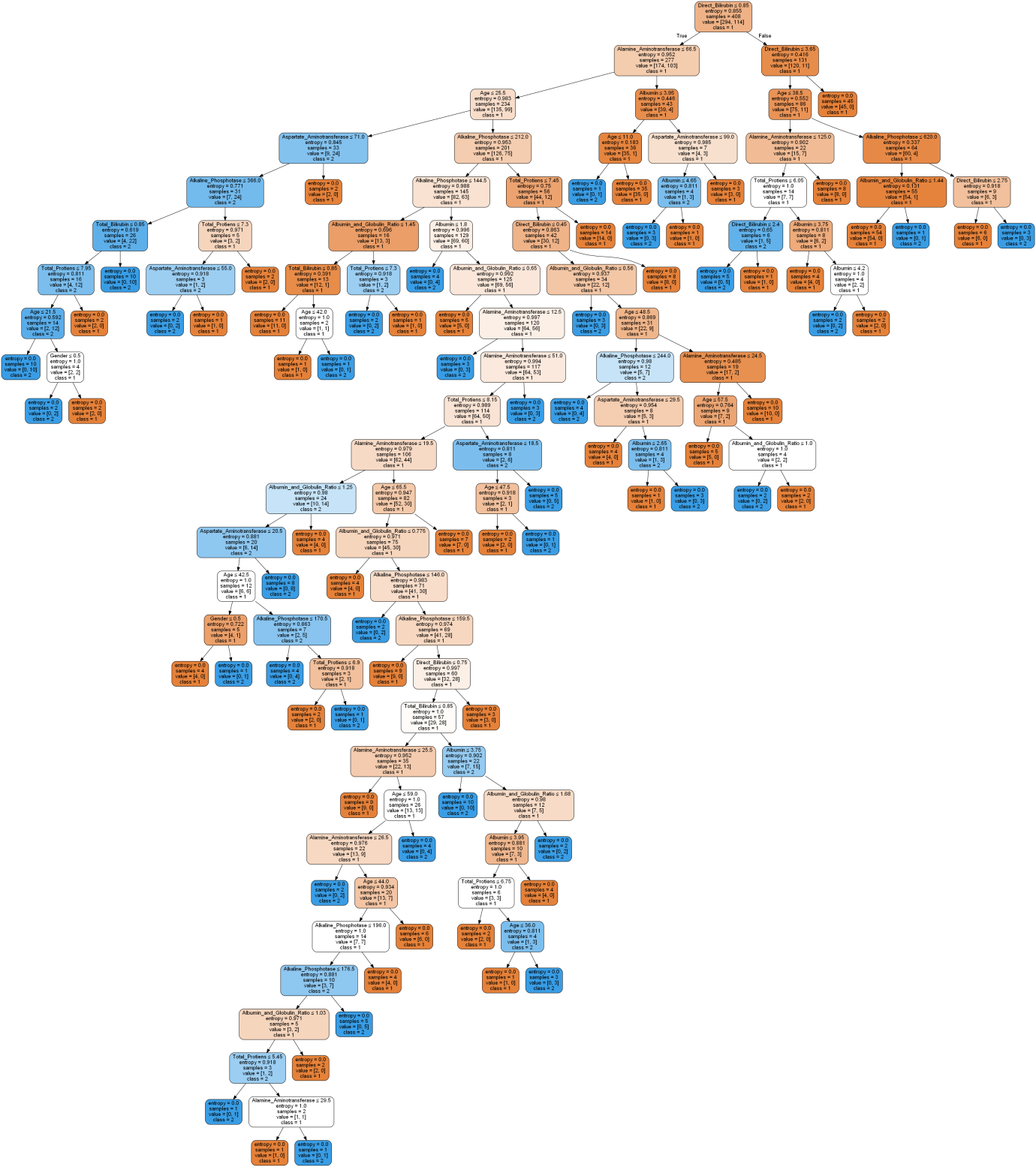
export\_graphviz(classifier, out\_file=dot\_data,

filled=True, rounded=True,

special\_characters=True,feature\_names = col\_names,class\_names=['1','2'])

graph = pydotplus.graph\_from\_dot\_data(dot\_data.getvalue()) graph.write\_png('liver\_disease20.png')

Image(graph.create\_png())

****

6.Pruning the decision tree by specifying the maximum depth

#Calculating the accuracy for max\_depth=3

x=0.0000000000

for i in range(0,30):

classifier= DecisionTreeClassifier(criterion="entropy",max\_depth=3)

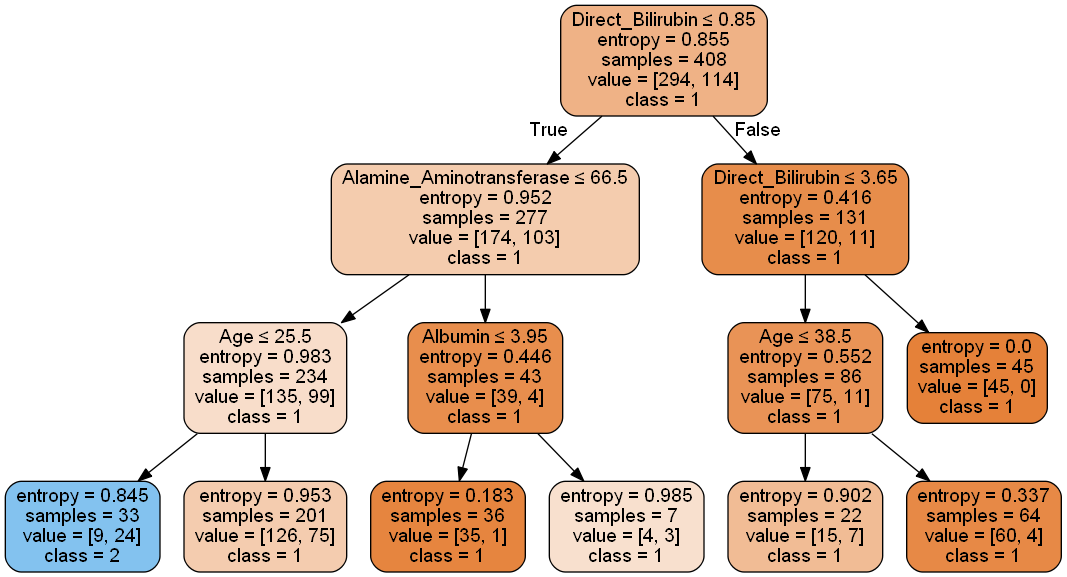
classifier.fit(x\_train, y\_train)

y\_pred = classifier.predict(x\_test)

x=x+metrics.accuracy\_score(y\_test, y\_pred)

print("Accuracy:",i," :",metrics.accuracy\_score(y\_test, y\_pred))

print("Average accuracy of 30 runs:",x/30)



**Result**

The result of the proposed work is obtained by implementing two different algorithms.The judgement of the results was done by comparing the accuracy of the two models.

**Logistic Regression**

from sklearn.metrics import accuracy\_score

print('Accuracy: \n', accuracy\_score(Y\_test,log\_predicted)\*100)

The accuracy obtained by logistic regression is 78.76%

**ACCURACY =78.76%**

**Decision Tree**

**Accuracy of 30 runs before pruning :**

Accuracy: 0 : 0.64

Accuracy: 1 : 0.6685714285714286

Accuracy: 2 : 0.6514285714285715

Accuracy: 3 : 0.6514285714285715

Accuracy: 4 : 0.64

Accuracy: 5 : 0.6685714285714286

.

.

.

Accuracy: 28 : 0.6571428571428571

Accuracy: 29 : 0.64

Average accuracy of 30 runs: 0.6546666666666667

**ACCURACY =65.47%**

**Accuracy of 30 runs after pruning :**

Accuracy: 0 : 0.68

Accuracy: 1 : 0.68

Accuracy: 2 : 0.68

Accuracy: 3 : 0.68

Accuracy: 4 : 0.68

.

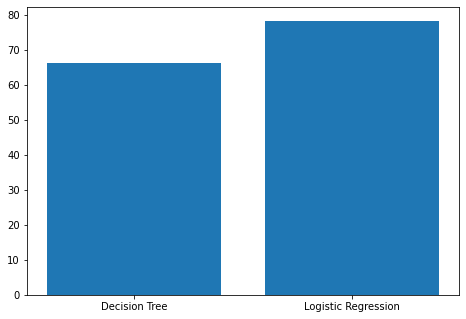
.

Accuracy: 28 : 0.68

Accuracy: 29 : 0.68

Average accuracy of 30 runs: 0.6799999999999998

**ACCURACY =68%**



Comparison of accuracy

|  |  |
| --- | --- |
| ALGORITHM | ACCURACY |
| Logistic Regression | 78.76% |
| Decision Tree | 68% |

**CONCLUSION**

In the proposed work, two different algorithms logistic regression and decision tree were implemented on liver disease dataset.Initially the dataset was explored. This is achieved by finding the null values if present and fill the null values by taking the mean,median or mode of that particular column ,transforming some particular columns such as target column and gender column using particular encoding techniques which would be helpful for classification algorithms.A small Exploratory Data Analysis was carried out such as number of patients suffering/not suffering from liver disease etc.Later the dataset is split for training and testing which is used by the algorithm .

After training and testing the model we find that the best result was given by logistic regression as compared to decision tree. Finally a liver disease prediction web application is created and deployed which can be used to check if a person is suffering from liver disease or not.

# **REFERENCES**

1. <https://archive.ics.uci.edu/>
2. <https://flask.palletsprojects.com/>
3. <https://seaborn.pydata.org/>
4. <https://scikit-learn.org/>
5. <https://www.javatpoint.com/machine-learning>
6. <https://scikit-learn.org/stable/modules/tree.html#tree>
7. <https://towardsdatascience.com/under-the-hood-decision-tree-454f8581684e>

**SNAPSHOTS OF THE PROJECT**

