SUMMER INTERNSHIP PROJECT REPORT

LIVER DISEASE PREDICTION

TEAM MEMBERS

AVNISH KUMAR KASHYAP (CSJMA16001390015)

GAURAV KUMAR YADAV (CSJMA16001390023)

DHEERAJ KUMAR GUPTA (CSJMA16001390022)

SUDHANSHU RAWAT (CSJMA16001390057)

RISHABH SINGH (CSJMA16001390044)

Table of Contents

Introduction

In India, delayed diagnosis of diseases is a fundamental problem due to a shortage of medical professionals. A typical scenario, prevalent mostly in rural and somewhat in urban areas is:

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 reports back to the hospital, where they are examined and the disease is identified.

The aim of this project is to somewhat reduce the time delay caused due to the unnecessary back and forth shuttling between the hospital and the pathology lab. Historically, work has been done in identifying the onset of diseases like heart disease, Parkinson's from various features, for example in this paper https://link.springer.com/chapter/10.1007/978-3-319-11933-5_17.In this case, a machine learning algorithm will be trained to predict a liver disease in patients.

Problem Statement

The problem statement is formally defined as:

'Given a dataset containing various attributes of 583 Indian patients, use the features available in the dataset and define a supervised classification algorithm which can identify whether a person is suffering from liver disease or not.'

The dataset for this problem is the ILPD (Indian Liver Patient Dataset) taken from the UCI Machine Learning Repository. Number of instances are 583. It is a multivariate data set, contain 10 variables that are age, gender, total Bilirubin, direct Bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT and Alkphos. All values are real integers. This data set contains 416 liver patient records and 167 non- liver patient records. The data set was collected from north east of Andhra Pradesh, India. This data set contains 441 male patient records and 142 female patient records. Any patient whose age exceeded 89 is listed as being of age "90".

Strategy

This seems to be a classic example of supervised learning. We have been provided with a fixed number of features for each data point, and our aim will be to train a variety of Supervised Learning algorithms on this data, so that, when a new data point arises, our best performing classifier can be used to categorize the data point as a positive example or negative. Exact details of the number and types of algorithms used for training are included in the 'Algorithms and Techniques' sub-section of the 'Analysis' part.

Metrics

In problems of disease classification like this one, simply comparing the accuracy, that is, the ratio of correct predictions to total predictions is not enough. This is because depending on the context like severity of disease, sometimes it is more important that an algorithm does not wrongly predict a disease as a non-disease, while predicting a healthy person as diseased will attract a comparatively less severe penalty.

Thus, here we will use **F-beta score** as a performance metric, which is basically the weighted harmonic mean of precision and recall. Precision and Recall is defined as:

Precision=TP/ (TP+FP), Recall=TP/ (TP+FN), where TP=True Positive

FP=False Positive

FN=False Negative

In the same vein, F-beta score is:

F-beta score = $(1+\beta^2)$*precision*recall/ ((β^2*precision) +recall)

β = A number that decides relative weightage of precision and recall. In this case, a disease being classified as a non-disease will incur a high penalty. So, more emphasis is placed on recall.

Additionally, one more metric called as Receiver Operating Characteristics (ROC) curve will be used. It plots the curve of True Positive Rate vs. the False Positive Rate for a given algorithm, with a greater area under the curve indicating a better True Positive Rate for the same False Positive Rate, indicating the usefulness of the classifier.

Analysis

Exploring the Data

The ILPD dataset contains ten features as listed below:

1. Age
2. Gender
3. Total bilirubin
4. Direct bilirubin
5. Total proteins
6. Albumin
7. A/G ratio
8. SGPT
9. SGOT
10. Alkphos

TOTAL BILIRUBIN (TBil)

Normal values of total bilirubin range from 0.3–1.0 mg/dL.If bilirubin is not being attached to the glucose-derived acid (conjugated) in the liver or is not being adequately removed from the blood, it can mean that there is damage to your liver. Testing for bilirubin in the blood is therefore a good way of testing for liver damage.

DIRECT BILIRUBIN (DBil)

The reference range of direct bilirubin is 0.1-0.4 mg/dL.Bilirubin is a substance made when your body breaks down old red blood cells. This is a normal process. Direct bilirubin travels freely through your bloodstream to your liver. PROTEIN LEVEL: The normal range for total protein is between 6 and 8.3 grams per decilitre (g/dL). This range may vary slightly among laboratories. These ranges are also due to other factors such as: age.

ALBUMIN

It is the most abundant protein in human blood plasma; it constitutes about half of serum protein. It is produced in the liver. The reference range for albumin concentrations in serum is approximately 35 - 50 g/L (3.5 - 5.0 g/dL).

A/G RATIO

The albumin to globulin (A/G) ratio has been used as an index of disease state; however, it is not a specific marker for disease because it does not indicate which specific proteins are altered. The normal A/G ratio is 0.8-2.0

SGPT

An SGPT blood test is a test used to measure the amount of the enzyme glutamate pyruvate transaminase (GPT) in blood serum. This enzyme is found in much greater concentration in the liver. This test is also sometimes known as ALT or, where it is also combined with several other tests to find out how well the liver is functioning. The normal range of values SGPT is from 7 to 56 units per liter of serum.

SGOT

The SGOT test measures one of two liver enzymes, called AST, which stands for aspartate amino transferase. An SGOT test (or AST test) evaluates how much of the liver enzyme is in the blood. The normal range of values for AST (SGOT) is about 5 to 40 units per liter of serum (the liquid part of the blood).

ALKPHOS

An alkaline phosphatase (ALP) test is used measure the amount of the enzyme in your blood and help in diagnosing the problem. It checks how your liver is working. The normal range is 44 to 147 IU/L (international units per liter) or 0.73 to 2.45 microkat/L.

All features, except Gender are real valued integers. The last column, Disease, is the label (with '1' representing presence of disease and '2' representing absence of disease). Total number of data points is 583, with 416 liver patient records and 167 non liver patient records. A brief description of dataset, including parameters like mean, min, max for each column is given below:

	Age	Total Bilirubin	Direct Bilirubin	Total Proteins	Albumin	A/G ratio	SGPT	SGOT	Alkphos	Disease
count	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	579.000000	583.000000
mean	44.746141	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.141852	0.947064	1.286449
std	16.189833	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.795519	0.319592	0.452490
min	4.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.900000	0.300000	1.000000
25%	33.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.600000	NaN	1.000000
50%	45.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.100000	NaN	1.000000
75%	58.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.800000	NaN	2.000000
max	90.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.500000	2.800000	2.000000

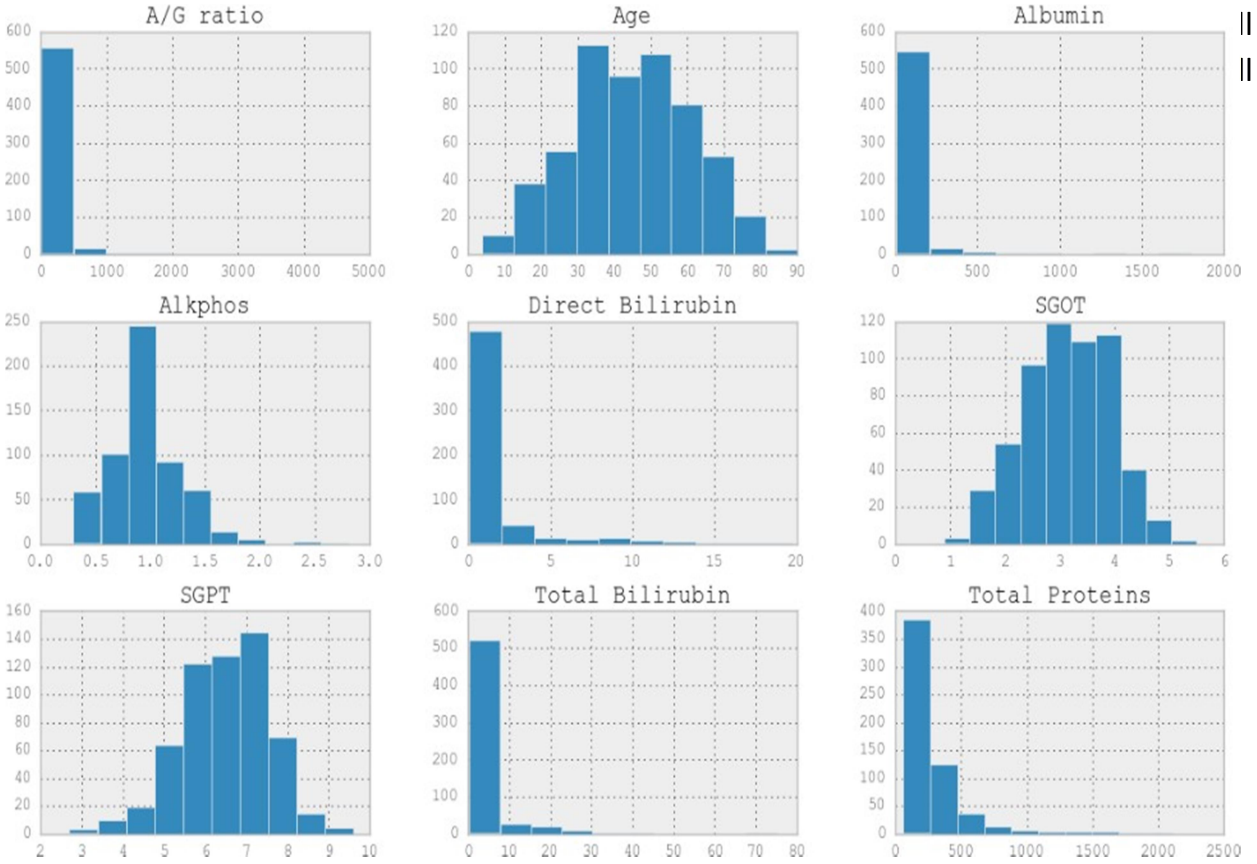
Fig 1 : Statistics of all the features of the model

	Age	Gender	Total Bilirubin	Direct Bilirubin	Total Proteins	Albumin	A/G ratio	SGPT	SGOT	Alkphos	Disease
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1

Fig 2 : A snapshot of the dataset containing first 5 rows

In the description of this dataset, it is observed that some values are 'Null' for the 'Álkphos' column. Accordingly, 4 rows containing those values are removed. Remaining number of rows in the dataset is 579.

Exploratory Visualization



Skewed features found are Albumin, Direct Bilirubin, A/G ratio, Total Bilirubin, Total Protein. On these, a log transformation is applied to reduce their range.

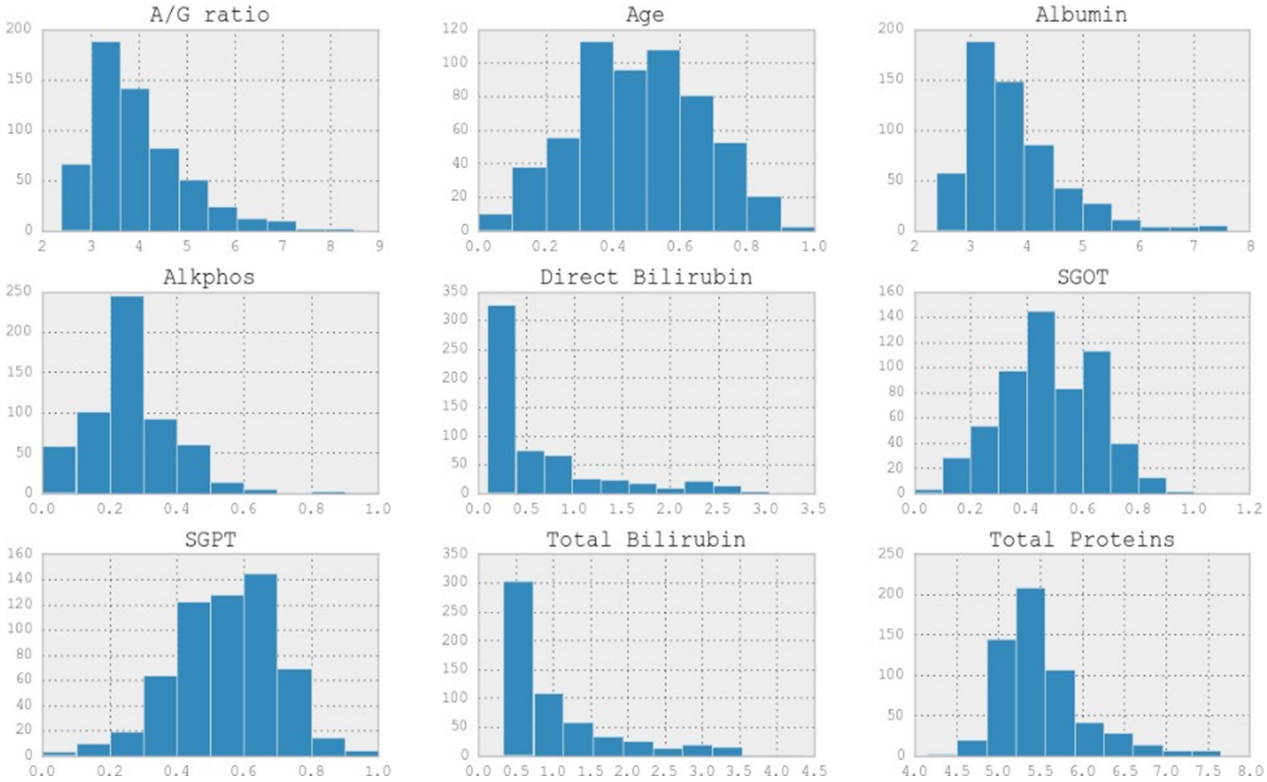


Fig 4: Histograms of all the transformed features

Algorithms and Techniques

Three supervised learning approaches are selected for this problem. Care is taken that all these approaches are fundamentally different from each other, so that we can cover as wide an umbrella as possible in term of possible approaches. For example- We will not select Random Forest and Ada Boost together as they come from the same family of 'ensemble' approaches. The choice of algorithms was influenced from these two sources:

http://scikit-learn.org/stable/tutorial/machine_learning_map

http://stackoverflow.com/questions/2595176/when-to-choose-which-machine-learning-classifier

For each algorithm, we will try out different values of a few hyperparameters to arrive at the best possible classifier. This will be carried out with the help of grid search cross validation technique. The algorithms are described below:

Aritifical Neural Networks

What are artificial neural networks (ANN)?

Human brains interpret the context of real-world situations in a way that computers can't. Neural networks were first developed in the 1950s to address this issue. An artificial neural network is an attempt to simulate the network of neurons that make up a human brain so that the computer will be able to learn things and make decisions in a humanlike manner. ANNs are created by programming regular computers to behave as though they are interconnected brain cells.

How do artificial neural networks work?

Artificial neural networks use different layers of mathematical processing to make sense of the information it's fed. Typically, an artificial neural network has anywhere from dozens to millions of artificial neurons—called units—arranged in a series of layers. The input layer receives various forms of information from the outside world. This is the data that the network aims to process or learn about. From the input unit, the data goes

through one or more hidden units. The hidden unit's job is to transform the input into something the output unit can use The majority of neural networks are fully connected from one layer to another. These connections are weighted; the higher the number the greater influence one unit has on another, similar to a human brain. As the data goes through each unit the network is learning more about the data. On the other side of the network is the output units, and this is where the network responds to the data that it was given and processed.

Cognitive neuroscientists have learned a tremendous amount about the human brain since computer scientists first attempted the original artificial neural network. One of the things they learned is that different parts of the brain are responsible for processing different aspects of information and these parts are arranged hierarchically. So, input comes into the brain and each level of neurons provide insight and then the information gets passed on to the next, more senior level. That's precisely the mechanism that ANNs are trying to replicate.

In order for ANNs to learn, they need to have a tremendous amount of information thrown at them called a training set. When you are trying to teach an ANN how to differentiate a cat from dog, the training set would provide thousands of images tagged as a dog so the network would begin to learn. Once it has been trained with the significant amount of data, it will try to classify future data based on what it thinks it's seeing (or hearing, depending on the data set) throughout the different units. During the training period, the machine's output is compared to the human- provided description of what should be observed. If they are the same, the machine is validated. If it's incorrect, it uses back propagation to adjust its learning—going back through the layers to tweak the mathematical equation. Known as deep learning, this is what makes a network intelligent.

Logistic Regression:

Since the outcome is binary and we have a reasonable number of examples at our disposal compared to number of features, this approach seems suitable. At the core of this method is a logistic or sigmoid function that quantifies the difference between each prediction and its corresponding true value. When presented with a number of inputs , it assigns different weights to features(based on their relative importance).Since for this data it already knows the output beforehand, it continuously adjusts the weights such that when these weights summed up with their features are introduced in the logistic function, the results are as near as possible to the actual ones. Once presented with a test value, it again inserts the value into our logistic function and returns the output as

a number between 0 and 1, which represents the probability of that test value being in a particular class.

Beating this benchmark model means that our method is suitable to be applied in the real world, as the problem dataset inherently favours Logistic Regression in terms of limited sample size and large number of positive examples (people having the disease). In real world, a much greater dataset size can be created due to the large population, and the percentage of positive cases will also be quite less. In such cases, the algorithms chosen are known to perform better, and our assumption of placing more emphasis on recall will also be better placed. So, if any one of these models manages to beat or even have comparable performance metrics to Logistic regression, it will have a high probability of giving a better performance in a real world scenario.

Gradient Descent:

Before talking about Stochastic Gradient Descent (SGD), Gradient Descent is a very popular optimization technique in Machine Learning and Deep Learning and it can be used with most, if not all, of the learning algorithms. A gradient is basically the slope of a function; the degree of change of a parameter with the amount of change in another parameter. Mathematically, it can be described as the partial derivatives of a set of parameters with respect to its inputs. The more the gradient, the steeper the slope. Gradient Descent is a convex function.
Gradient Descent can be described as an iterative method which is used to find the values of the parameters of a function that minimizes the cost function as much as possible. The parameters are initially defined a particular value and from that, Gradient Descent is run in an iterative fashion to find the optimal values of the parameters, using calculus, to find the minimum possible value of the given cost function.

Stochastic Gradient Descent (SGD):

The word '*stochastic*' means a system or a process that is linked with a random probability. Hence, in Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. In Gradient Descent, there is a term called "batch" which denotes the total number of samples from a dataset that is used for calculating the gradient for each iteration. In typical Gradient Descent optimization, like Batch Gradient Descent, the batch is taken to be the whole dataset. Although, using the whole dataset is really useful for getting to the minima in a less noisy or less random manner, but the problem arises when our datasets get really huge. Suppose, you have a million samples in your dataset, so if you use a typical Gradient

Descent optimization technique, you will have to use all of the one million samples for completing one iteration while performing the Gradient Descent, and it has to be done for every iteration until the minima is reached. Hence, it becomes computationally very expensive to perform.

This problem is solved by Stochastic Gradient Descent. In SGD, it uses only a single sample, i.e., a batch size of one, to perform each iteration. The sample is randomly shuffled and selected for performing the iteration.

Methodology

Data Preprocessing

As explained in the section 'Exploring the data', rows having 'Null' values were removed from the dataset. Thereafter, log transformation was applied to features which were showing a skewed pattern (Albumin, Direct Bilirubin, A/G ratio, Total Bilirubin, Total Protein).

Thereafter, all columns in the dataset except 'Gender' are normalized. We use MinMaxScaler here as StandardScaler gives very low values here, with some in the order of 10^-16, which might be difficult to relate to and visualize. Transformation is given by:

X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0)) X_scaled = X_std * (max - min) + min

Then we use pd.get_dummies() method to one-hot encode the feature 'Gender' as well as the label 'Disease' (with the integer '1' representing presence of disease).

Implementation

The dataset will be split into training and testing set as a 80-20 split using train_test_split method from sklearn. Random state will be specified as a particular number so that we have a means for comparison later, by specifying the same random state.

Before applying any supervised learning technique, we will implement a naïve predictor, that will simply return that every data point has 'Disease'= True. We will check our accuracy on that predictor. Note that in this case naive predictor will perform artificially well unlike in real world, a large proportion of patients (around 70%) do have the disease.

Then, a method called as 'train_predict' is defined that takes as input the following: learner, sample_size, X_train, y_train, X_test, y_test. It returns the accuracy and F-beta score on training and testing set respectively.

The three classifiers are sent to the 'train_predict' method, with 20%, 50% and 100% of training data respectively so that it can be seen how performance varies with sample size.

Performance Metrics of Artificial Neural Network

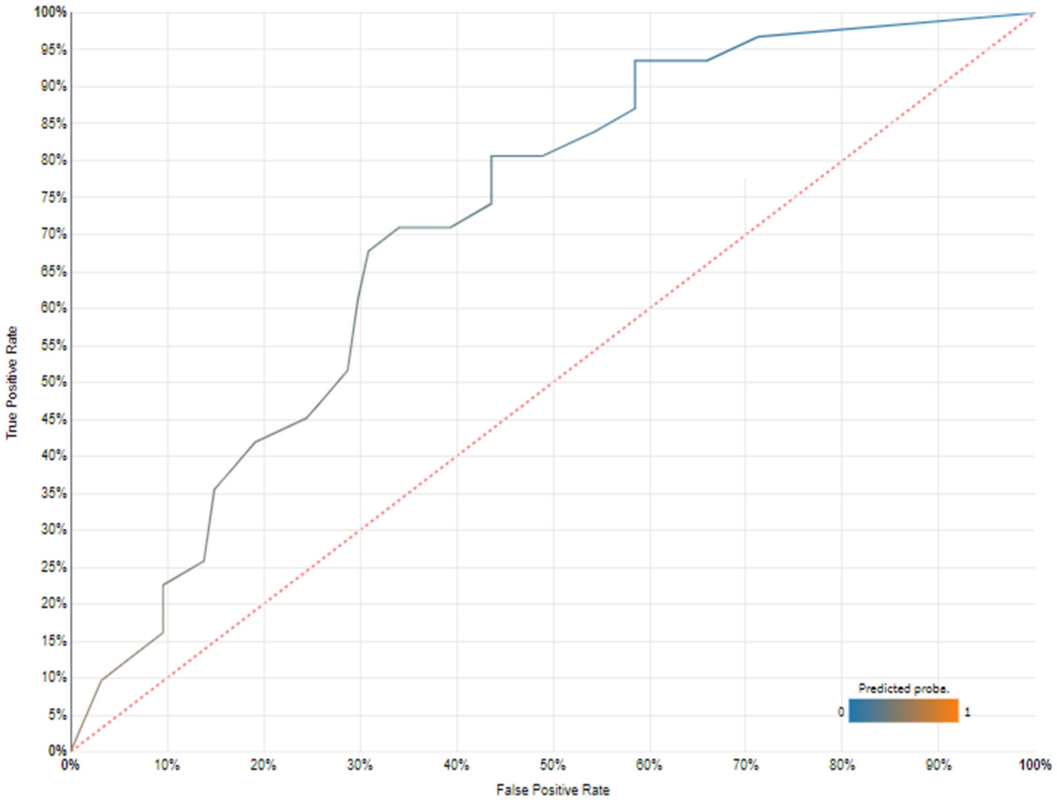


Fig :ROC Curve for ANN(AUC .717)

	Predicted 1	Predicted 0	Total
Actually 1	22	9	31
Actually 0	35	59	94
Total	57	68	125

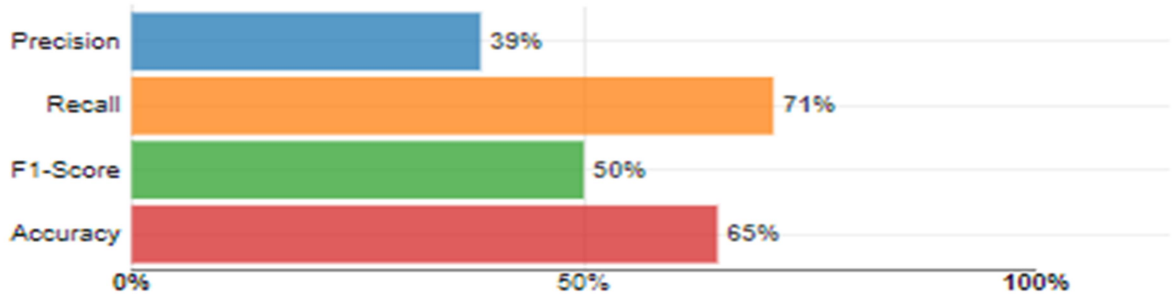


Fig : Confusion Matrix

Performance Metrics of Logistic Regression

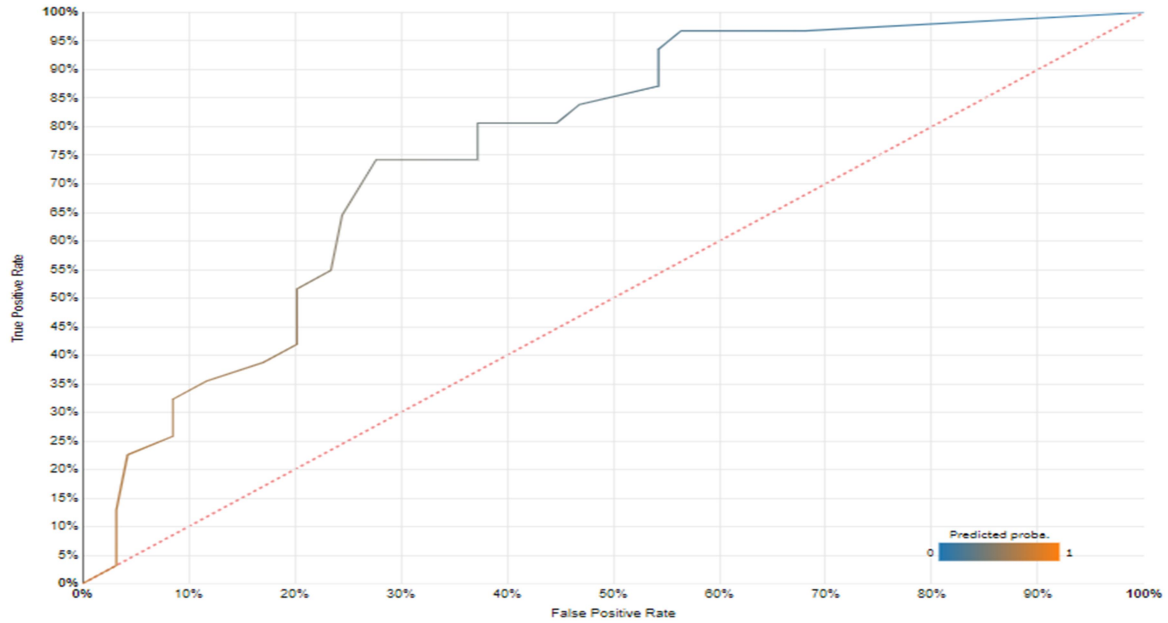


Fig : ROC Curve (AUC .766)

	Predicted 1	Predicted 0	Total
Actually 1	22	9	31
Actually 0	26	68	94
Total	48	77	125

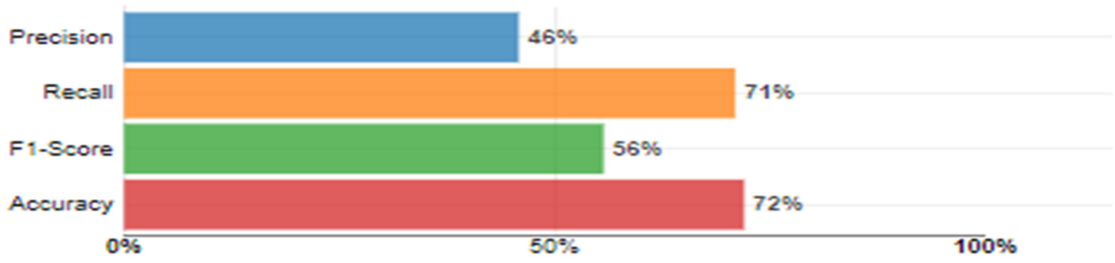


Fig : Confusion Matrix

Performance Metrics of Stochastic Gradient Descent(SGD)

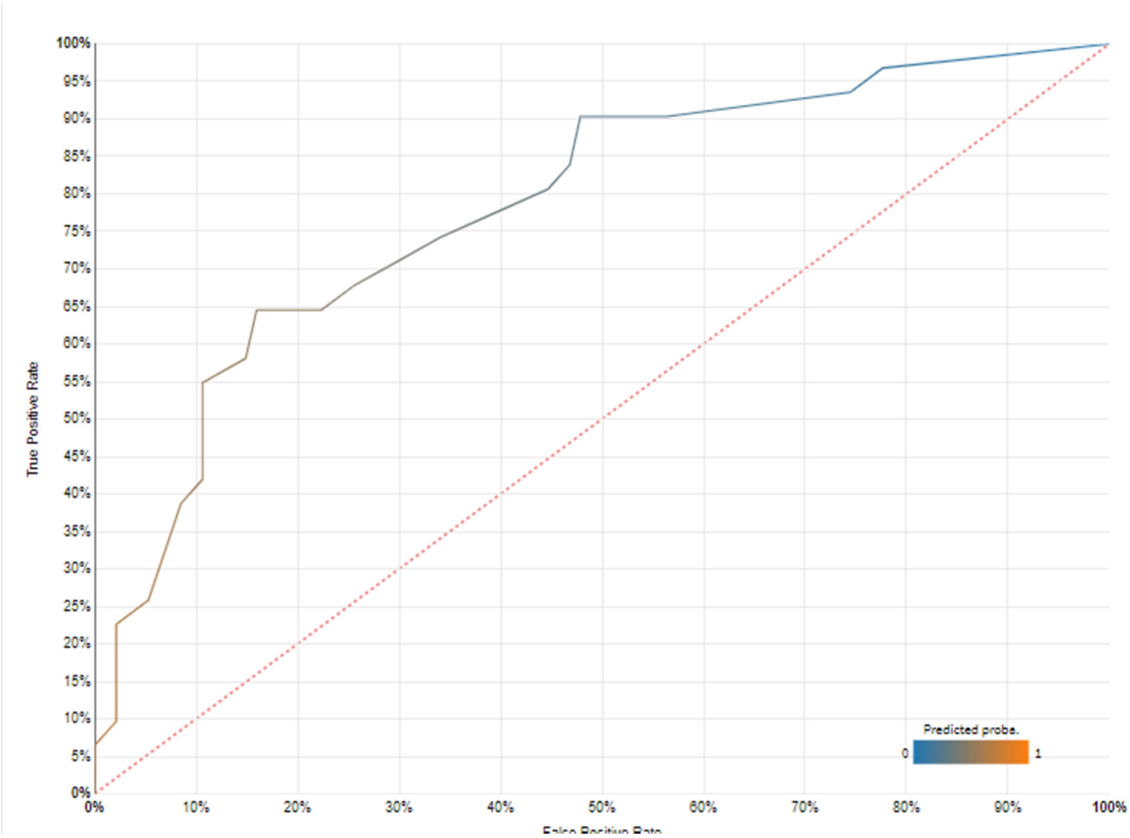


Fig : ROC Curve (AUC .793)

	Predicted 1	Predicted 0	Total
Actually 1	18	13	31
Actually 0	12	82	94
Total	30	95	125

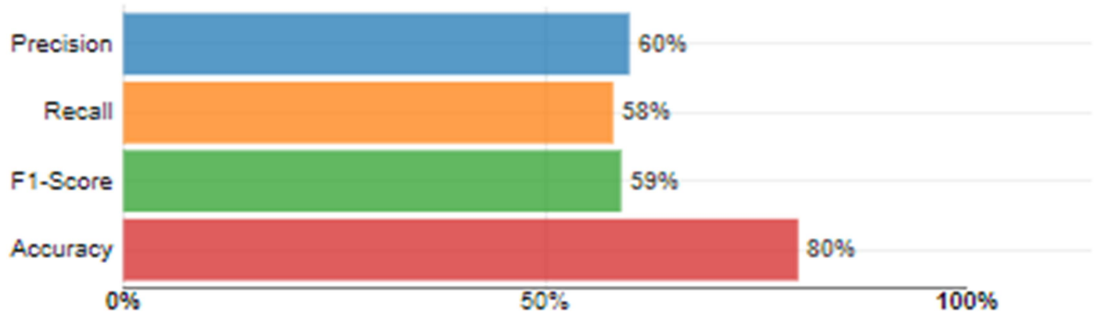


Fig : Confusion Matrix

Result

Since SGD performs best on all the four parameters choosen by us such as accuracy,f1-score,recall and precision.Also its area under curve for ROC curve is maximum.

Therefore, the SGD Model Selected for the analysis.

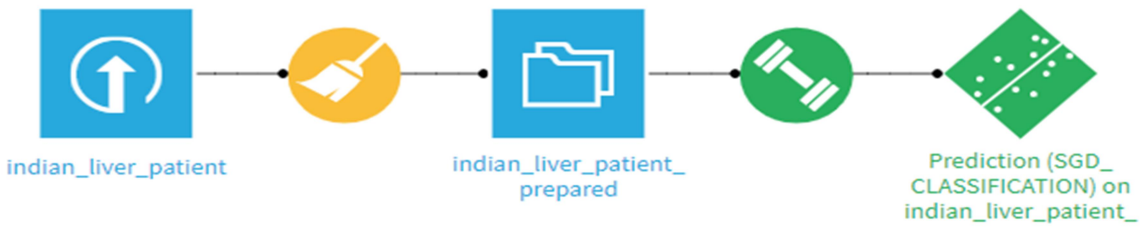
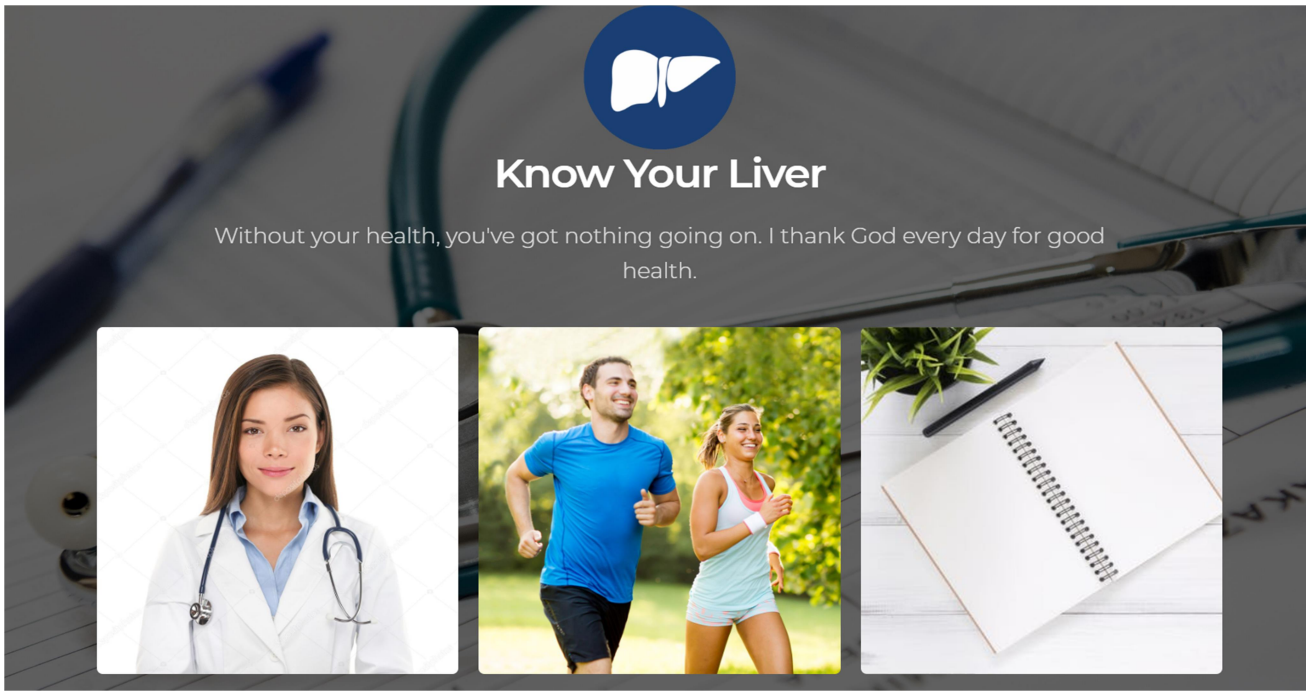
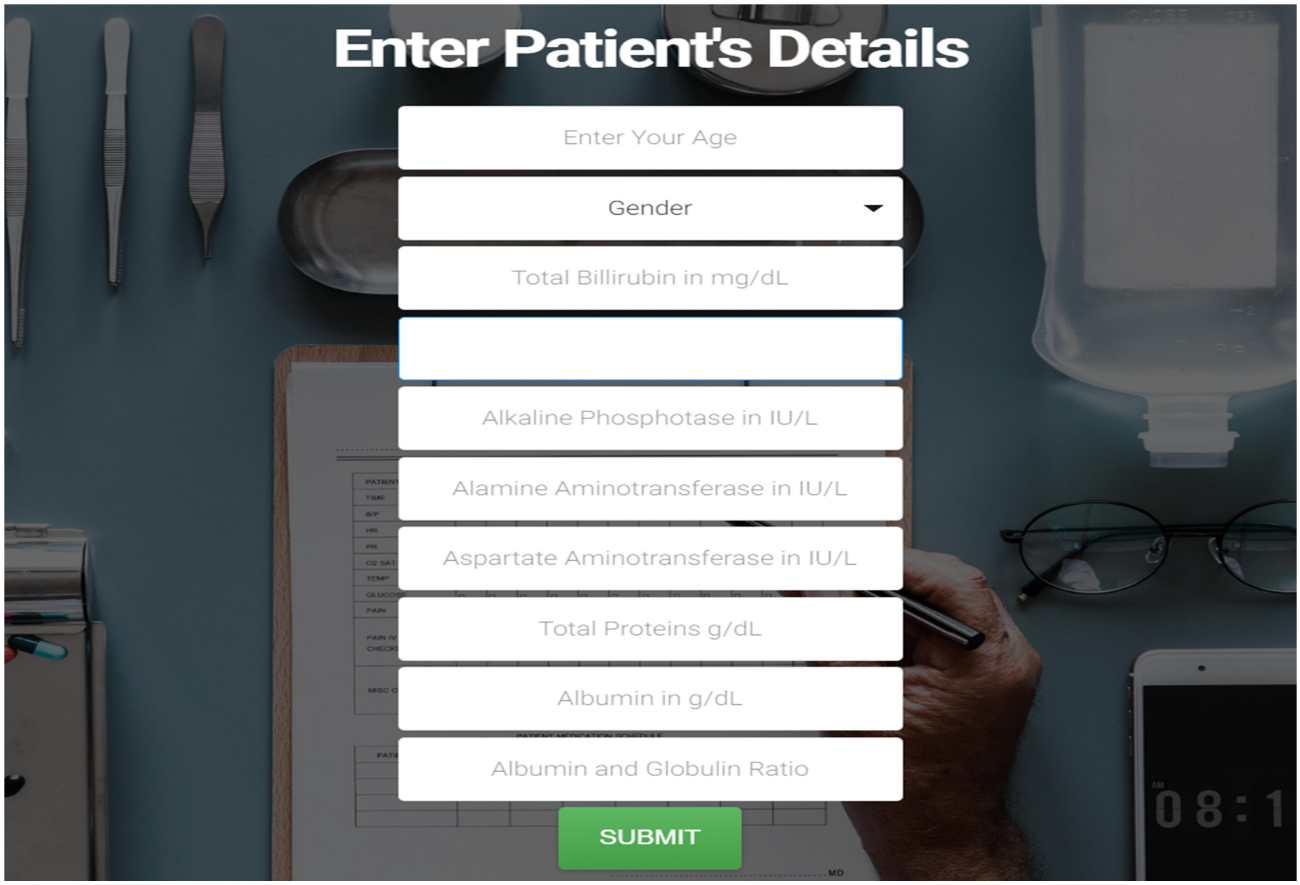


Fig : Flow Diagram of the process opted for the modle

Screenshots





Conclusion

A website is made for predicting the liver disease.This site is made possible because of the analysis we have performed on the liver disease data.It will help us to choose the write model for prediction.

Based on the analysis, we write the code for the model in python language.

The code saved the model on the system.

We also make an attractive web page and integrade it with our model.

Finally, we upload our model on the AWS cloud and it is accessible all over the world.

References

- This dataset was downloaded from the UCI ML Repository: *Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.*
- Data science Studio (Dataiku).
- http://scikit-learn.org/stable/tutorial/machine_learning_map
- http://stackoverflow.com/questions/2595176/when-to-choose-which-machine-learning-classifier