LIVER PATIENT ANALYSIS USING MACHINE LEARNING

1. INTRODUCTION

Overview

Liver diseases averts the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of liver disease at an early stage is a complex task for the doctors. The main objective of this project is to analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease.

This Project examines data from liver patients concentrating on relationships between a key list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease. Here we are building a model by applying various machine learning algorithms find the best accurate model. And integrate to flask based web application. User can predict the disease by entering parameters in the web application.

Proposed System

Developing a machine that will enhance in the diagnosis of the disease will be of a great advantage in the medical field. These systems will help the physicians in making accurate decisions on patients and also with the help of Automatic classification tools for liver diseases (probably mobile enabled or web enabled), one can reduce the patient queue at the liver experts such as endocrinologists.

2. LITERATURE SURVEY

2.1 Existing Problem

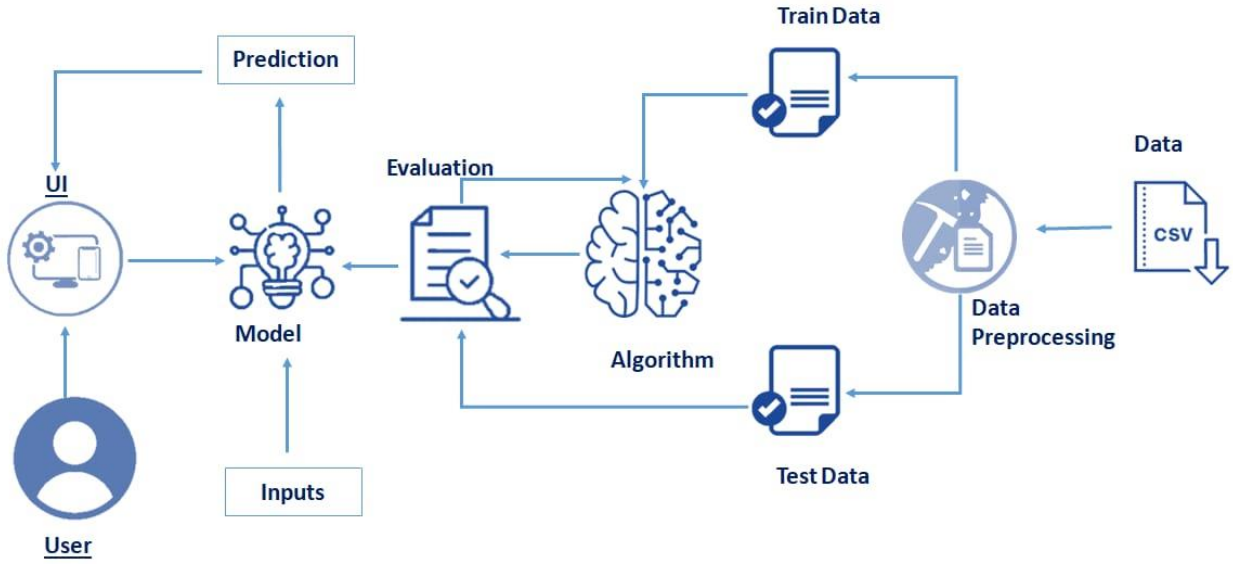
Problems with liver patients are not easily discovered in an early stage as it will be functioning normally even when it is partially damaged. An early diagnosis of liver problems will increase patient’s survival rate. Liver failures are at high rate of risk among Indians. It is expected that by 2025 India may become the World Capital for Liver Diseases. The widespread occurrence of liver infection in India is contributed due to deskbound lifestyle, increased alcohol consumption and smoking. There are about 100 types of liver infections.

2.2 Proposed system

Developing a machine that will enhance in the diagnosis of the disease will be of a great advantage in the medical field. These systems will help the physicians inmaking accurate decisions on patients and also with the help of Automatic classification tools for liver diseases (probably mobile enabled or web enabled), one can reduce the patient queue at the liver experts such as endocrinologists.

3. THEORETICAL ANALYSIS

3.1 Block Diagram

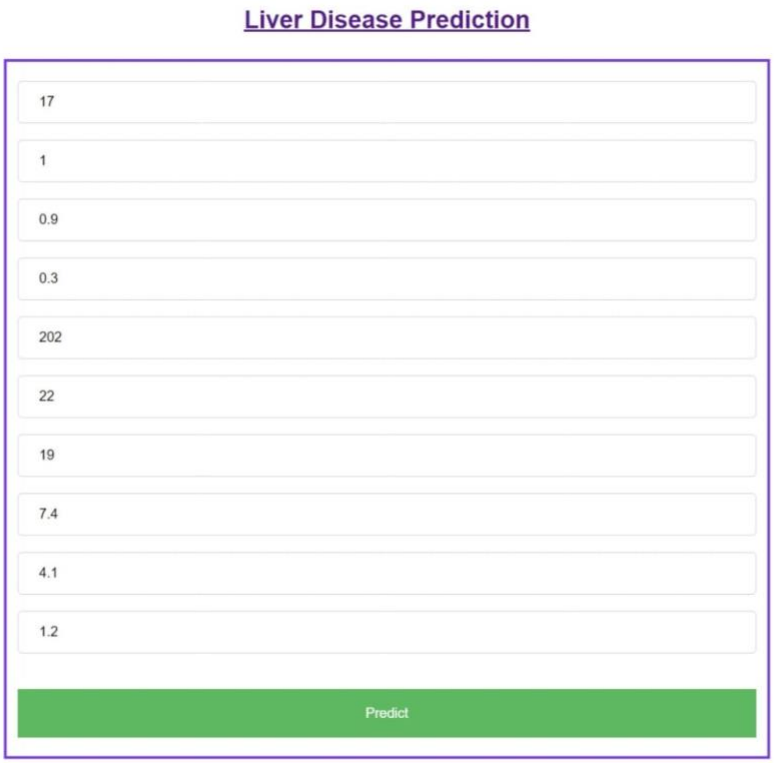


3.2 Hardware/Software Designing
*Hardware requirements
• 2 GB ram or above
• Dual core processor or above
• Internet connection
*Software requirements
• Anaconda Navigator
• Python packages

4. EXPERIMENTAL INVESTIGATION
The main objective of this research is to use classification algorithms to identify the liver patients from healthy individuals. In this study, FOUR classification algorithms Logistic Regression, Support Vector Machines (SVM), K Nearest Neighbor (KNN) and artificial neural networks (ANN) have been considered for comparing their performance based on the liver patient data. Further, the model with the highest accuracy is implemented as a user friendly Graphical User Interface (GUI) using Tkinter package in python. The GUI can be readily utilized by doctors and medical practitioners as a screening tool for liver disease.
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 patient data.
5. FLOWCHART
• User interacts with the UI (User Interface) to type their message as input.
• Message input is analysed by the model which is integrated.
• Once model analyses the given message, the prediction is showcased on the UI
1. Data Collection
a. Collect the dataset or create the dataset
2. Understanding the data

a. Importing the required libraries
b. Reading the Dataset
c. EDA on Dataset
d. Take care of missing data
e. Data Visualization
f. Cleaning The Text
g. Building count vectors with scikit-learn Count-Vectorizer for text classification
h. Splitting Data into Train and Test
3. Model Building
a. Training and testing the model
b. Evaluation of Model
c. Saving the model
4. Application Building
a. Create an HTML file
b. Build Python Code
5. Final UI
a. Dashboard Of the flask app

6. RESULT
Results 1.



Results:

No Worries!!! You don't have Liver Disease.

Result 2.

Liver Disease Prediction

78

0

1.7

0.7

160

61

29

6.1

1.6

0.96

Predict

Results:

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).

7. ADVANTAGES AND DISADVANTAGES

Advantages

1. No medical expertise required: You don't need to have any knowledge of medical science and liver diseases to predict the liver disease using this application. All you need to do is enter the details being asked, which are already present in the blood test report(some like age, gender are already known) and then you will get the results of prediction.

2.High accuracy: The system predicts the results with 100 % accuracy for the dataset that we have used while creating this application. While the accuracy might be different in some cases, it will still be high enough to be trustworthy at a large scale.

3.Immediate results: The results here are predicted within seconds of entering the details. You don't need to wait for a doctor to come, unlike in traditional method.

8. APPLICATION

• Liver patient analysis and disease prediction

9. CONCLUSION

Diseases related to liver and heart are becoming more and more common with time. With continuous technological advancements, these are only going to increase in the future. Although people are becoming more conscious of health nowadays and are joining yoga classes, dance classes; still the sedentary lifestyle and luxuries that are continuously being introduced and enhanced; the problem is going to last long. So, in such a scenario, our project will be extremely helpful to the society. With the dataset that we used for this project, we got 100 % accuracy for SVM model, and though it might be difficult to get such accuracies with very large datasets, from this project's results, one can clearly conclude that we can predict the risk of liver diseases with accuracy of 90 % or more.

10. FUTURE SCOPE

Today almost everybody above the age of 12 years has smartphones with them, and so we can incorporate these solutions into an android app or ios app. Also it can be incorporated into a website and these app and website will be highly beneficial for a large section of society.

11. BIBLIOGRAPHY

• https://www.kaggle.com/code/sanjames/liver-patients-analysis-prediction-accuracy
•
• https://www.ijert.org/liver-disease-prediction-system-using-machine-learning-techniques

Biomarkers for prediction of liver fibrosis in patients with chronic alcoholic liver disease written by Sylvie Naveau and Bruno Runyard.

• Strategic analysis in prediction of liver disease using classification algorithms written by Piyush Kr Shukla andBinish Khan.

• Software based prediction of liver disease with feature selection and classification techniques witten Jagdeep Singh, Sandeep Bagga and Ranjodh Kaur.

• Liver disease prediction using SVM and Naïve Bayes algorithm written by S Dhayanand.

12. APPENDIX

```
From google.colab import drive
Drive.mount('/content/drive')
Import numpy as np
Import pandas as pd
Import matplotlib.pyplot as plt
Import seaborn as sns
Dataset = pd.read_csv('/content/drive/MyDrive/Indian Liver Patient Records.zip')
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(dat
aset['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
#spliting data into tringdata 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 1st 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 accuracy_score, precision_score
Print(accuracy_score(y_test,ann_y_pred))
Print(precision_score(y_test,ann_y_pred))
Import pickle
Pickle.dump(knn_classifier,open('model.pkl','wb'))
Pickle.dump(sc, open('sc.pkl','wb'))
```