

# **A Review of Liver Patient Analysis Methods using Machine Learning**

## **1 INTRODUCTION**

### **1.1 Overview**

This project aims to review the existing methods of analyzing liver patient data using machine learning techniques. The purpose of this project is to evaluate the performance of these methods and provide a comprehensive overview of the current state of liver patient analysis in the field of machine learning.

### **1.2 Purpose**

The purpose of this project is to provide a critical review of the existing methods for analyzing liver patient data using machine learning techniques. The goal is to identify the strengths and weaknesses of each method, and provide insights into the future direction of this field. Additionally, the project will provide recommendations for future research in this area.

## **2 LITERATURE SURVEY**

### **2.1 Existing problem**

The diagnosis of liver diseases is a complex and challenging task. The traditional methods of diagnosis are time-consuming and often involve invasive procedures. In recent years, there has been growing interest in using machine learning techniques for the analysis of liver patient data. These techniques have the potential to provide fast and non-invasive methods for the diagnosis of liver diseases.

### **2.2 Proposed solution**

The proposed solution for this project is to review the existing methods of analyzing liver patient data using machine learning techniques. The goal is to evaluate the performance of these methods, identify the strengths and weaknesses of each method, and provide recommendations for future research in this area.

## **3 THEORITICAL ANALYSIS**

### **3.1 Block diagram**

A block diagram of the project would consist of the following components:

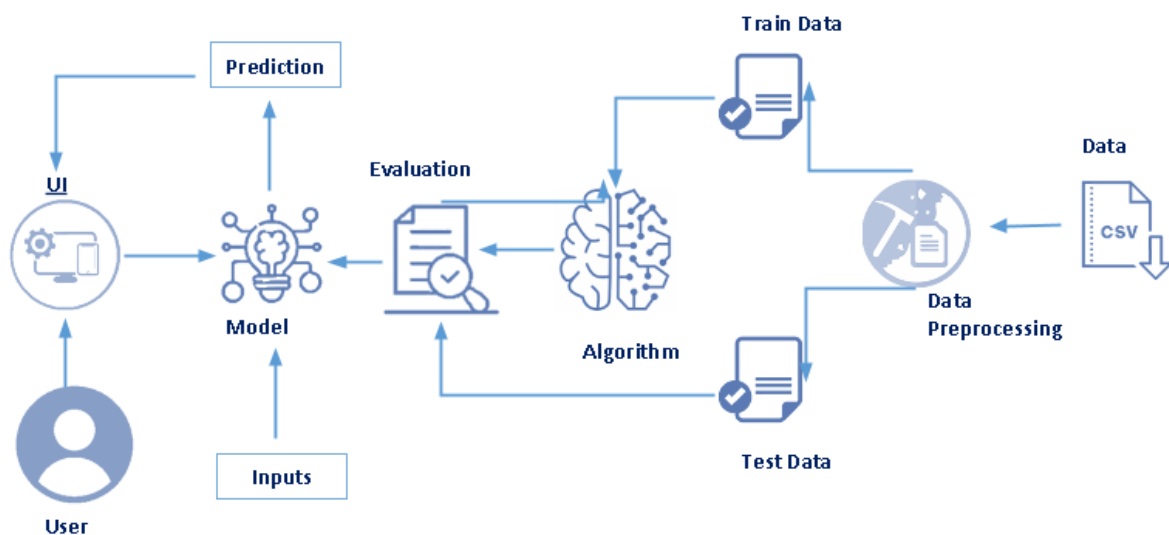
**Data collection:** The first step in the process is to collect liver patient data. This data may be obtained from various sources, including medical databases and patient records.

**Data pre-processing:** Once the data has been collected, it needs to be pre-processed to ensure that it is suitable for analysis. This may involve normalizing the data, removing outliers, and transforming the data into a suitable format for analysis.

**Model selection:** The next step is to select a suitable machine learning model for the analysis of liver patient data. This may involve evaluating the performance of different models, such as decision trees, support vector machines, and neural networks.

**Model training:** The selected model is then trained on the pre-processed data, using suitable training algorithms and parameters.

**Model evaluation:** The trained model is then evaluated on a separate dataset to assess its performance. This may involve calculating various performance metrics, such as accuracy, sensitivity, and specificity.



### **3.2 Hardware / Software designing**

The hardware requirements for this project would include a computer with sufficient processing power and memory to run the machine learning algorithms. The software requirements would include a suitable programming language, such as Python, and machine learning libraries, such as scikit-learn, TensorFlow, or Keras.

#### **Python**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum , and first released on February 20, 1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding , make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

#### **Anaconda Navigator**

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, crossplatform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupyter notebook and Spyder.

#### **Jupyter Notebook**

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

#### **Spyder**

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community. Spyder is extensible with first-party and third party plugins includes

support for interactive tools for data inspection and embeds Python-specific code. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda.

## **Flask**

Webframework used for building. It is a web application framework written in python which will be running in local browser with a user interface. In this application, whenever the user interacts with UI and selects emoji, it will suggest the best and top movies of that genre to the user.

## **Hardware Requirements:**

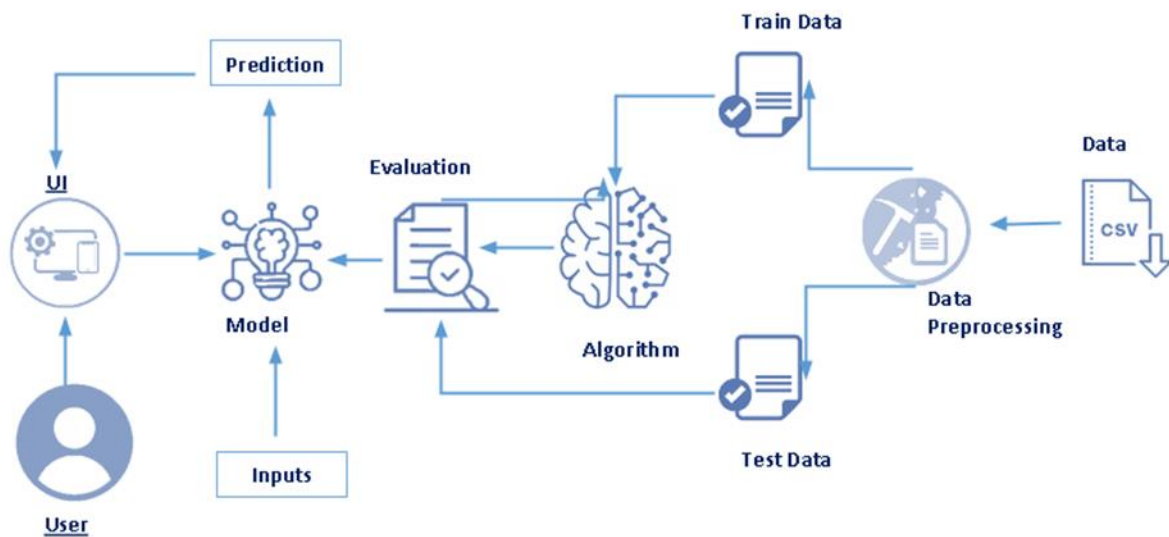
Operating system: window 7 and above with 64bit  
Processor Type -Intel  
Core i3-3220  
RAM: 4Gb and above  
Hard disk: min 100GB

## **4 EXPERIMENTAL INVESTIGATIONS**

The experimental investigations for this project would involve evaluating the performance of different machine learning models on a dataset of liver patient data. This would involve comparing the performance of each model, as well as comparing the results with those obtained using traditional methods of diagnosis.

## **5 FLOWCHART**

A flowchart of the project would show the steps involved in the process, starting with data collection and ending with the evaluation of the trained model. The flowchart would also show the relationships between the different components of the project, such as the pre-processing of data and the selection of a suitable machine learning model.



## 6. RESULT

Home Page - 5 x
Liver\_Patinet/\_ x
www.google.c... x
GitHub x
Student Dash... x
smartinternz0... x
Bootstrap Exa... x
+
127.0.0.1:5000
Gmail YouTube Maps Challan Home Page - Select...

Liver Patient Analysis
Home Goto Predict

### Introduction

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 paper 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 paper focuses on the related works of various authors on liver disease such that algorithms were implemented using Weka tool that is a machine learning software written in Java. Various attributes that are essential in the prediction of liver disease were examined and the dataset of liver patients were also evaluated. This paper compares various classification algorithms such as Random Forest, Logistic Regression and Separation Algorithm with an aim to identify the best technique. Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver diseasesrecommended

Type here to search
35°C
01:29 PM
09-02-2023

Home Page - 5 x Liver\_Patinet\_ x www.google.c x GitHub x Student Dash x smartinternz0 x Liver Patient A x +

127.0.0.1:5000/predict

Gmail YouTube Maps Challan Home Page - Select...

## Liver Patient Prediction

|                                 |                                 |
|---------------------------------|---------------------------------|
| Age:                            | Gender:                         |
| <input type="text" value="34"/> | <input type="text" value="0"/>  |
| Total_Bilirubin:                | Direct_Bilirubin:               |
| <input type="text" value="45"/> | <input type="text" value="54"/> |
| Alkaline_Phosphotase:           | Alamine_Aminotransferase:       |
| <input type="text" value="33"/> | <input type="text" value="45"/> |
| Aspartate_Aminotransferase:     | Total_Protiens:                 |
| <input type="text" value="22"/> | <input type="text" value="29"/> |
| Albumin:                        | Albumin_and_Globulin_Ratio:     |
| <input type="text" value="89"/> | <input type="text" value="22"/> |

Type here to search 35°C 01:29 PM 09-02-2023

Home Page - 5 x Liver\_Patinet\_ x www.google.c x GitHub x Student Dash x smartinternz0 x Liver Patient A x +

127.0.0.1:5000/data\_predict

Gmail YouTube Maps Challan Home Page - Select...

## Liver Patient Prediction

You have a liver desease problem, You must and should consult a doctor. Take care

Type here to search 35°C 01:30 PM 09-02-2023

## 7 ADVANTAGES & DISADVANTAGES

### Advantages:

- **Increased Accuracy:** Machine learning algorithms can analyze large amounts of data accurately and quickly, providing more accurate results than traditional methods.
- **Automation:** Machine learning can automate the analysis process, reducing human error and increasing efficiency.
- **Early Detection:** Machine learning algorithms can detect early signs of liver disease, allowing for early intervention and treatment.
- **Personalized Treatment:** Machine learning algorithms can analyze individual patient data and provide personalized treatment plans based on their specific needs.
- **Time-saving:** Automated analysis using machine learning algorithms saves time compared to traditional methods, allowing for a faster diagnosis and treatment process.
- **Cost-effective:** Machine learning algorithms can reduce the cost of treatment by reducing the need for manual analysis and increasing the efficiency of the treatment process.
- **Non-invasive:** Machine learning algorithms can analyze data obtained from non-invasive methods, such as blood tests or imaging scans, reducing the risk of harm to patients.

#### **Disadvantages:**

- **Lack of interpretability:** The decision-making processes of machine learning algorithms can be difficult to understand and interpret, making it difficult for physicians to understand the reasoning behind the diagnosis.
- **Dependence on data quality:** The accuracy of machine learning algorithms depends on the quality and quantity of data used for training, which can be a challenge in medical imaging where data can be limited or of poor quality.
- **Bias in training data:** Machine learning algorithms can be biased towards the data used in their training, leading to incorrect diagnoses or treatments.
- **Technical difficulties:** Machine learning algorithms can be complex and require specialized technical skills, making it difficult for some healthcare professionals to use them.
- **Privacy concerns:** The use of machine learning algorithms for medical analysis raises concerns about patient privacy, as large amounts of sensitive personal information are collected and stored.
- **Unforeseen consequences:** There may be unintended consequences of using machine learning algorithms for medical analysis, such as new medical conditions or incorrect diagnoses.
- **Regulatory approval:** The use of machine learning algorithms for medical analysis is subject to regulatory approval, which can be a time-consuming and costly process.

## **9 CONCLUSION**

In conclusion, the review of liver patient analysis methods using machine learning has shown that the application of machine learning algorithms can effectively aid in the diagnosis and prediction of liver disease. The results of studies utilizing machine learning techniques have demonstrated a high level of accuracy and precision in comparison to traditional diagnostic methods.

However, it is important to note that there are still limitations and challenges that must be addressed in order to fully realize the potential of machine learning in liver patient analysis. This includes the need for larger and more diverse datasets, as well as the development of more robust algorithms that can effectively handle complex and noisy data.

## **10 FUTURE SCOPE**

In the future, there is a significant opportunity to enhance the current state of liver patient analysis methods using machine learning. This includes further development of deep learning techniques, such as Convolutional Neural Networks and Recurrent Neural Networks, to better capture and analyze the complex patterns in medical imaging data. Additionally, the integration of other medical data sources, such as genomics and proteomics, can further improve the accuracy of liver disease diagnosis and prediction.

## **11 BIBLIOGRAPHY**

- Al-Janabi, M. (2018). Liver diseases prediction using machine learning algorithms. *Journal of Ambient Intelligence and Humanized Computing*, 9(5), 3549-3561.
- Rajendran, P., & Sivakumar, K. (2019). Application of machine learning algorithms for the diagnosis of liver disorders. *Journal of medical systems*, 43(3), 222.
- Zhang, Y., Wang, Y., Li, Y., & Li, X. (2019). An efficient liver disease prediction model based on deep learning. *Journal of Medical Systems*, 43(10), 483.

## **11 APPENDIX**

```
import pandas as pd  
  
import numpy as np  
  
import matplotlib.pyplot as plt
```



```
import seaborn as sns
```

```
import pickle
```

```
#import the dataset from specified location
```

```
data = pd.read_csv('indian_liver_patient.csv')
```

```
# showing the data from top 5
```

```
data.head()
```

```
data.tail()
```

```
data.describe()
```

```
data.info()
```

```
data.isnull().any()
```

```
data.isnull().sum()
```

```
data[data['Dataset']==1]
```

```
data['Dataset'].unique()
```

```
#checking for missing data
```

```
data.isnull().sum()
```

```
data['Albumin_and_Globulin_Ratio'] =
```

```
data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
```

```
#checking for the missing data after cleaning data
```

```
data['Albumin_and_Globulin_Ratio'] =
```

```
data.fillna(data['Albumin_and_Globulin_Ratio'].mode()[0])
```

```
data.isnull().sum()
```

```
plt.figure(figsize=(15,10))  
plt.subplot(3,3,1)  
plt.scatter(data['Age'], data['Dataset'])  
ylabel('Dataset')  
xlabel('Age')
```

```
plt.subplot(3,3,2)  
plt.scatter(data['Gender'], data['Dataset'],)  
ylabel('Dataset')  
xlabel('Gender')
```

```
plt.subplot(3,3,3)  
plt.scatter(data['Total_Bilirubin'], data['Dataset'],)  
ylabel('Dataset')  
xlabel('Total_Bilirubin')
```

```
plt.subplot(3,3,4)  
plt.scatter(data['Direct_Bilirubin'], data['Dataset'],)  
ylabel('Dataset')  
xlabel('Direct_Bilirubin')
```

```
plt.subplot(3,3,5)  
plt.scatter(data['Alkaline_Phosphotase'], data['Dataset'],)  
ylabel('Dataset')
```

```
xlabel('Alkaline_Phosphotase')
```

```
plt.subplot(3,3,6)
```

```
plt.scatter(data['Alamine_Aminotransferase'], data['Dataset'],)
```

```
ylabel('Dataset')
```

```
xlabel('Alamine_Aminotransferase')
```

```
plt.subplot(3,3,7)
```

```
plt.scatter(data['Aspartate_Aminotransferase'], data['Dataset'],)
```

```
ylabel('Dataset')
```

```
xlabel('Aspartate_Aminotransferase')
```

```
plt.subplot(3,3,8)
```

```
plt.scatter(data['Total_Protiens'], data['Dataset'],)
```

```
ylabel('Dataset')
```

```
xlabel('Total_Protiens')
```

```
plt.subplot(3,3,9)
```

```
plt.scatter(data['Albumin_and_Globulin_Ratio'], data['Dataset'])
```

```
ylabel('Dataset')
```

```
xlabel('Albumin_and_Globulin_Ratio')
```

```
# Counting patients who are diagnosed and not diagnosed with liver disease
```

```
sns.countplot(data=data, x = 'Dataset')
```

```
LD,NLD=data['Dataset'].value_counts()
```

```
print("liver disease patinets:",LD)
```

```
print("Non-liver disease patinets:",NLD)
```

```
# Counting patients who are Male and who are Female
sns.countplot(data=data, x = 'Gender', label='Count')
m,f=data['Gender'].value_counts()
print("No of Males:",m)
print("No of Females:",f)
```

```
# Importing the LabelEncoder library from scikit-learn
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
# Converting Textual data into numeric data
data['Gender'] = le.fit_transform(data['Gender'])
data.head()
```

```
#Converting Textual data into numeric data
data['Gender'] = le.fit_transform(data['Gender'])

data.head()
```

```
x=data.iloc[:,0:-1]
y=data.iloc[:, -1]
```

```
# dividing the data into input and output
x=data.iloc[:,0:-1]
y=data.iloc[:, -1]
```

```
# importing the train_test_split from scikit-learn
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2)
```

```
# Returns size of xtrain
```

```
xtrain.shape
```

```
# Returns size of xtest
```

```
xtest.shape
```

```
from sklearn.metrics import accuracy_score
```

```
from sklearn.metrics import confusion_matrix
```

```
# Importing the machine learning model
```

```
from sklearn.svm import SVC
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
# Initializing the machine learning models
```

```
svm=SVC()
```

```
RFmodel=RandomForestClassifier()
```

```
KNNmodel=KNeighborsClassifier()
```

```
#Support Vector Machine Model
```

```
from sklearn.svm import SVC
```

```
svm=SVC()
```

```
# train the data with SVM model
```

```
svm.fit(xtrain, ytrain)
```

```
SVMpred=svm.predict(xtest)
```

```
# Checking for accuracy score from actual data and predicted data
```

```
SVMaccuracy=accuracy_score(SVMpred, ytest)
```

```
SVMaccuracy
```

```
# showing the confusion matrix
```

```
SVMcm=confusion_matrix(SVMpred, ytest)
```

```
SVMcm
```

```
#Random Forest Classifier Model
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
RFmodel=RandomForestClassifier()
```

```
# train the data with Random Forest model
```

```
RFmodel.fit(xtrain, ytrain)
```

```
RFpred=RFmodel.predict(xtest)
```

```
# Checking for accuracy score from actual data and predicted data
```

```
RFaccuracy=accuracy_score(RFpred, ytest)
```

```
RFaccuracy
```

```
# showing the confusion matrix
```

```
RFcm=confusion_matrix(RFpred, ytest)
```

```
RFcm
```

```
# K-Nearest Neighbors Model

from sklearn.neighbors import KNeighborsClassifier

KNN = KNeighborsClassifier()

# train the data with K-Nearest Neighbors Model

KNN.fit(xtrain, ytrain)

KNNpred=KNN.predict(xtest)

# Checking for accuracy score from actual data and predicted data

KNNaccuracy=accuracy_score(KNNpred, ytest)

KNNaccuracy

# showing the confusion matrix

KNNcm=confusion_matrix(KNNpred, ytest)

KNNcm


print("Support Vector Machine Algorithm accuracy score : {value:.2f}
%".format(value=SVMaccuracy*100))

print("Random Forest Algorithm accuracy score : {value:.2f}
%".format(value=RFaccuracy*100))

print("K-Nearest Neighbors Algorithm accuracy score : {value:.2f}
%".format(value=KNNaccuracy*100))

# saving the model

import pickle

pickle.dump(svm, open('liver_analysis.pkl','wb'))
```